CANMET

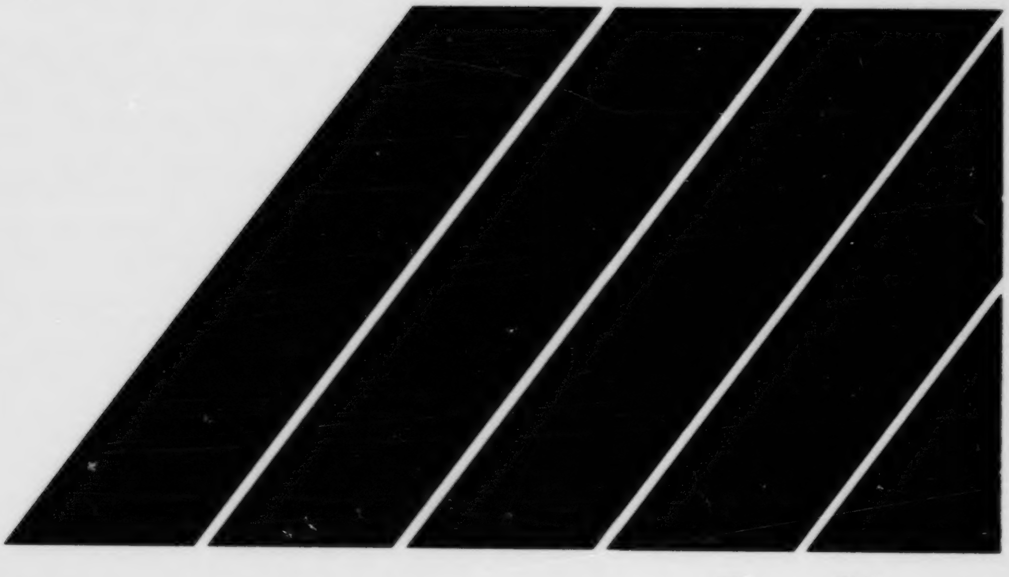
Canada Centre for
Mineral and Energy
Technology

Centre canadien de la
technologie des
minéraux et de l'énergie

**Efficiency
and Alternative
Energy
Technology
Branch**

**Direction de la
technologie de
l'efficacité énergétique
et des énergies
de remplacement**

MEASUREMENT AND COMPUTER SIMULATION OF HEAT TRANSFER IN GLAZING SYSTEMS



Energy, Mines and
Resources Canada

Énergie, Mines et
Ressources Canada

Canada

THE ENERGY OF OUR RESOURCES - THE POWER OF OUR IDEAS

L'ENERGIE DE NOS RESSOURCES - NOTRE FORCE CREATRICE

[redacted]

91-08136/

MEASUREMENT AND COMPUTER SIMULATION OF HEAT TRANSFER IN GLAZING SYSTEMS

PREPARED FOR:

Efficiency and Alternative Energy Technology Branch
Energy, Mines and Resources Canada
Ottawa, Ont.
DSS Contract No. 23280-6-6122
September, 1990

PREPARED BY:

Advanced Glazing System Laboratory
Department of Mechanical Engineering
University of Waterloo
Waterloo, Ont.
N2L 3G1

SCIENTIFIC AUTHORITY:

Joel Allarie
Efficiency and Alternative Energy Technology Branch
Energy, Mines and Resources Canada
580 Booth Street
Ottawa, Ont.
K1A 0E4

265 pp

MICROMEDIA

CITATION

Wright, J.L., Advanced Glazing System Laboratory, University of Waterloo. Measurement and Computer Simulation of Heat Transfer in Glazing Systems. DSS Contract No. 23280-6-6122. Efficiency and Alternative Energy Technology Branch, Energy, Mines and Resources Canada, Ottawa, Ontario. 1990 (155 pp.)

Copies of this report may be obtained through the following:

Efficiency and Alternative Energy Technology Branch
Energy, Mines and Resources Canada
580 Booth Street, 7th Floor
Ottawa, Ont.
K1A 0E4

or

Document Delivery Service
Library and Documentation Services Division
CANMET
Energy, Mines and Resources Canada
562 Booth Street
Ottawa, Ont.
K1A 0G1

DISCLAIMERS

This report is distributed for informational purposes only and does not necessarily reflect the views of the Government of Canada nor constitute an endorsement of any commercial product or person. Neither Canada nor its ministers, officers, employees or agents makes any warranty in respect to this report or assumes any liability arising out of this report.

The Waterloo Research Institute, University of Waterloo assumes no liabilities for the use of, or damages resulting from the use of, any information, apparatus, method or process resulting from the work described in this report.

NOTE

Funding for this project was provided by the Federal Panel on Energy Research and Development, Energy, Mines and Resources Canada.

© Minister of Supply & Services Canada 1991
Catalogue No. M91-7/150-1991E
ISBN. 0-662-18969-8

ACKNOWLEDGEMENTS

This work was supported by the Federal Panel on Energy R&D (PERD)

Appreciation is extended to Professor Harry Sullivan, for his time and effort and for providing the opportunity to undertake this research project.

The valuable contributions of Prof. Gord Stubley, for his day-to-day assistance with numerical models, and Prof. George Raithby, for suggestions leading to the successful formulation of a perturbation scheme, are also acknowledged.

Guarded heater plate experiments were performed using the equipment of the Solar Thermal Research Laboratory (STRL) at the University of Waterloo, Department of Mechanical Engineering. The assistance of Prof. K.G.T. Hollands, by providing equipment and giving time for discussion, is appreciated.

The cooperation of Mr. Michael Glover (Edgetech Ltd., Ottawa), Mr. Werner Lichtenberger and Mr. Gordon Crix (Tremco Ltd., Toronto) and Mr. James Larsen (Cardinal I.G., Minneapolis) is gratefully acknowledged. Their help in obtaining test samples was instrumental in the completion of this work.

Thanks are offered to Jeff Baker for his help in the laboratory and with the VISION code.

ABSTRACT

Existing methods can be used to calculate heat loss through glazing systems and window frames with reasonable accuracy but they do not account for the top-to-bottom differences in heat transfer caused by fill gas motion. They cannot be expected to predict the minimum indoor pane temperature that governs the onset of condensation on windows exposed to low outdoor temperatures. The current research was undertaken in order to firmly establish the idea that the minimum indoor pane temperature is not a function of edge-seal conduction alone but also depends upon fill gas convection. Toward this end a numerical model for predicting heat flux and temperature distributions in glazing systems was developed. This model allows for the detailed 2-D simulation of conductive, radiative and convective heat transfer. Major steps required to successfully complete the model included the measurement of edge-seal thermal resistance for a variety of edge-seal designs and the development of a 2-D model for natural convection of a gas in a vertical, rectangular cavity of high aspect ratio.

Simulation results were directly compared with measurements (taken using a guarded heater plate apparatus) at every stage of the investigation. In all cases discrepancies were small. This result lends credibility to both the simulation procedures and the guarded heater plate experiments.

Development of the 2-D numerical model for natural convection is noteworthy because it is capable of simulating secondary cells in the fill gas flow and because agreement with experimental data was extremely good. This model involves a very simple formulation unlike previous models capable of resolving secondary cells.

The measured thermal resistance values of edge-seals provide a direct comparison between various edge-seal designs and offer a significant insight into the mechanisms governing heat transfer in edge-seals. The literature contains no other edge-seal thermal resistance measurement results.

RÉSUMÉ

Les méthodes existantes peuvent être utilisées pour calculer assez exactement la perte de chaleur à travers les vitrages et les cadres de fenêtres, mais elles ne tiennent pas compte des écarts de transfert de chaleur entre le haut et le bas dus au mouvement du gaz de remplissage. On ne peut espérer qu'elles prévoient la température minimale de la vitre intérieure à partir de laquelle il y a condensation sur les vitrages exposés à de faibles températures extérieures. Le recherche actuelle a été entreprise pour confirmer que cette température minimale n'est pas uniquement fontion de la conduction par le joint périphérique, mais aussi de la convection du gaz de remplissage. A cette fin, un modèle numérique de prévision de la distribution des flux thermiques et de la température dans les vitrages a été mis au point. Ce modèle permet une simulation 2D détaillée des transferts de chaleur par conduction, rayonnement et convection. Les principales étapes nécessaires pour assurer le succès total du modèle comprennent la mesure de la résistance thermique du joint périphérique selon les types et l'élaboration d'un modèle 2D de la convection naturelle d'un gaz dans une cavité rectangulaire verticale très allongée.

Les résultats de la simulation ont été directement comparés aux mesures (prises à l'aide d'une plaque chauffante protégée) à chaque étape de l'étude. Dans tous les cas, les écarts étaient faibles. On peut donc conclure que la méthode de simulation et les expériences avec la plaque chauffante sont faibles.

Le modèle 2D élaboré pour la convection naturelle est remarquable parce qu'il permet de simuler les cellules secondaires dans la circulation du gaz de remplissage et que les résultats sont en accord étroit avec les données expérimentales. Ce modèle est beaucoup plus simple que les modèles de résolution des cellules secondaires antérieurs.

Les mesures de résistance thermique des joints périphériques permettent de comparer directement les divers types de joints et de bien comprendre les mécanismes régissant le transfert de chaleur dans les joints périphériques. La documentation ne fait état d'aucun autre résultat de mesures de la résistance thermique des joints périphériques.

TABLE OF CONTENTS

ABSTRACT ii

RÉSUMÉ iii

LIST OF TABLES vi

LIST OF ILLUSTRATIONS vii

INTRODUCTION 1

  REFERENCES 5

BACKGROUND 9

  ASHRAE 9
  DEFINITION OF WINDOW AREAS 9
  DESIGN CONDITIONS 9
  U-VALUE 10
  SHADING COEFFICIENT 11
  NET ENERGY GAIN OR LOSS 12
  CURRENT GLAZING SYSTEM DESIGN PRACTICE 13
  REFERENCES 15

GLAZING SYSTEM ANALYSIS 16

  VISION 1-D GLAZING SYSTEM ANALYSIS 16
  EDGE-GLASS ANALYSIS 20
  1-D PREHEAT/SUPPLY AIR GLAZING SYSTEM MODEL 21
  REFERENCES 22

CENTRE-GLASS U-VALUE MEASUREMENT USING THE UNIVERSITY OF WATERLOO NATURAL CONVECTION APPARATUS 28

  INTRODUCTION 28
  GLAZING SYSTEM HEAT TRANSFER MEASUREMENT 31
    Adaptation of Apparatus 31
    Measurement Procedure 31
  MEASUREMENT RESULTS 35
  DISCUSSION 39
  REFERENCES 40

THERMAL RESISTANCE MEASUREMENT OF GLAZING SYSTEM EDGE-SEALS 49

  INTRODUCTION 49
  TEST PROCEDURE 50
  RESULTS 53
  DISCUSSION 54
  CONCLUSIONS 57
  REFERENCES 58

NATURAL CONVECTION IN A VERTICAL RECTANGULAR SLOT 63

REVIEW 63
Background 63
Edge-Glass Heat Transfer 64
Natural Convection Between Glazings 65
Discussion 73
2-D NUMERICAL MODEL FOR NATURAL CONVECTION 73
Formulation 73
Grid Sizing 77
Perturbation 77
Solution Procedure 80
Calculation of Nusselt Number 81
Test for Convergence of the Coefficient Loop 83
Results 84
DISCUSSION 87
REFERENCES 89

2-D NUMERICAL MODEL
FOR GLAZING SYSTEM THERMAL ANALYSIS 121

APPROACH 121
SOLID SECTIONS 121
THERMAL RADIATION MODEL 123
SOLUTION CONVERGENCE 127
PRELIMINARY RESULTS 128
COMPARISON WITH MEASURED HEAT FLUX 128
HEAT FLUX IN THE GLAZING CAVITY 134
TEMPERATURE PROFILES AT THE WARM GLAZING 136
DISCUSSION 137
REFERENCES 138

CONCLUSIONS 153

Appendix A 156
Wright, J.L., "Effective U-Values and Shading Coefficients of Preheat/Supply Air Glazing Systems", Proceedings of the 1986 SESCI Conference, Winnipeg, Canada, pp. 219-224.

Appendix B 163
Sample IGU-2D Input File;
Glazing System No. 3, Table 7.1

Appendix C 166
Sample IGU-2D Output File;
Results for Glazing System No. 3, Table 7.1

Appendix D 170
IGU-2D Fortran Source Code

LIST OF TABLES

2.1: ASHRAE Design Conditions 10

3.1 Capabilities of the VISION Glazing System Analysis Program 17

4.1 Comparison of Simulated and Measured C-Values; Metal Solar Control Coatings 38

4.2 Comparison of Simulated and Measured C-Values; Low-E Coating 39

5.1 List of Edge-Seal Test Samples 51

5.2 Summary of Results for Edge-Seal Test Units 53

6.1 Nu vs. Ra Simulation Results, A=40 85

6.2 Meter-Area Nusselt Numbers - Calculation vs. Measurements of ElSherbiny, A=40 88

6.3 Meter-Area Nusselt Numbers - Calculation vs. Measurements of Shewen, A=40 89

7.1 Calculated and Measured Glazing System Heat Flux Results 130

LIST OF ILLUSTRATIONS

3.1 1-D Glazing System Thermal Analysis Geometry 26

3.2 Sample Graphic Output from VISION 27

4.1 Sketch of the Hot Plate 43

4.2 Heater Plate/Heat Flux Meter Detail 43

4.3 Copper Plates with Triple Glazed Test Unit 44

4.4 Sample HFMV Strip Chart Recorder Output 45

4.5 Sample Test Results and Qe vs. HFMV Plot 46

4.6 Cross Section at Edge of Double Glazed Test Unit 47

4.7 VISION Graphic Output Showing Centre-Glass Simulation Results for Glazing System GS3 48

5.1 Typical Edge-Seal Design 59

5.2 Cross-Section of Edge-Seal Test Sample 59

5.3 The Corrugated Strip Edge-Seal Design 60

5.4 Cross-Section of Edge-Seal Test Sample with Foam Spacer 60

5.5 Measured Linear Conductance Results 61

5.6 Homogeneous Edge-Seal Dimensions 61

5.7a Examples of Homogeneous Edge-Seal Configurations, Equal Linear Conductance Values 62

5.7b Examples of Homogeneous Edge-Seal Configurations, Unequal Linear Conductance Values 62

6.1 Linear k-values Used to Estimate Edge-Seal Heat Loss 100

6.2 Plot Showing Isotherms: Standard Double Glazed Window, Wood Frame, Aluminum Spacer 101

6.3 Problem Domain for the Analysis of Natural Convection in a Vertical Slot 102

6.4 Nu vs. Ra and A Data of ElSherbiny, Raithby and Hollands for Air in a Vertical Cavity 103

6.5 Computed Mid-Height Velocity and Temperature Profiles of ElSherbiny, Fath and Refai, Vertical Air Layer, A=1 104

6.6 Laminar Flow Regimes 105

6.7 Sketch of Secondary and Tertiary Flow 106

6.8 Comparison of Predictions of Raithby and Wong with Experimental Data 107

6.9 Numerical Streamline Solutions of Lee and Korpela Showing Secondary Convective Cells in Air, A=20 108

6.10 Comparison of Predictions of Lee and Korpela with Experimental Data 109

6.11 Local Nusselt Number, Nu(y) for A=20 Based on Data from Korpela, Lee and Drummond 110

6.12 Convective Flow Regimes for Vertical Window Cavities 111

6.13 The Uniform, Staggered Grid 112

6.14 Calculation of Fluxes at Faces in the Finite Volume Computational Molecule 113

6.15 Solution Algorithm Flow Chart 114

6.16 Comparison of Available Simulation and Measured Nu vs. Ra Results for the Vertical Cavity, A=40 115

6.17 Stream Function Plots at Ra=6,800 and Ra=10,100 Line Spacing = 0.02 m^3/sec, A=40 116

6.18 Nu(y) versus y and Averaged Over Metering Areas of ElSherbiny, Ra=6,800, A=40 117

6.19 Nu(y) versus y and Averaged Over Metering Areas of Shewen, Ra=10,100, A=40 118

6.20 PATRAN Fringe Plots of T, u, v at Bottom End of Vertical Slot, Ra=6,800 and A=40 119

6.21 PATRAN Fringe Plots of T, u, v at Bottom End of Vertical Slot, Ra=10,100 and A=40 120

7.1 2-D Glazing System Thermal Analysis Geometry 139

7.2 Surface Temperature and Heat Flux at the Cavity Wall 140

7.3 Example of Geometry for Calculation of E_{p-nb} Radiative Exchange Factor 141

7.4 Calculated Heat Flux Distribution at Hot Boundary and Calculated Average Heat Flux for Heater Plate Areas (unit #3) 142

7.5a Calculated and Measured Heat Flux Results for Top, Middle and Bottom Guarded Heater Plate Areas (units 1, 2, 3 and 4) 143

7.5b Calculated and Measured Heat Flux Results for Top, Middle and Bottom Guarded Heater Plate Areas (units 5, 6 and 7) 144

7.6 Heat Flux Component Distributions Along Vertical Cut at Hot Cavity Wall (unit #1) 145

7.7 Heat Flux Component Distributions Along Vertical Cut at Hot Cavity Wall (unit #2) 146

7.8 Heat Flux Component Distributions Along Vertical Cut at Hot Cavity Wall (unit #3) 147

7.9 Heat Flux Component Distributions Along Vertical Cut at Hot Cavity Wall (unit #7) 148

7.10 Temperature Distribution Along Vertical Cut at Hot Cavity Wall (unit #1) 149

7.11 Temperature Distribution Along Vertical Cut at Hot Cavity Wall (unit #2) 150

7.12 Temperature Distribution Along Vertical Cut at Hot Cavity Wall (unit #3) 151

7.13 Temperature Distribution Along Vertical Cut at Hot Cavity Wall (unit #7) 152

Chapter 1

INTRODUCTION

As building construction methods progress toward higher levels of insulation, lower infiltration rates and larger areas of glazed aperture it becomes increasingly apparent that conventional window design is a weak link in building envelope performance. It has been estimated (1) that in the U.S. 1.7 million barrels of oil per day (5% of the national energy consumption) is tied to the thermal performance of windows.

In addition to providing only a minimum level of thermal resistance, conventional windows can create comfort problems, low allowable humidity levels and damaging accumulations of condensation. These shortcomings strengthen the desire for windows with high thermal resistance - especially in countries with cold winters such as Canada.

The use of double glazing started around the time of the American Civil War and is still the most common method of providing a reasonable level of thermal resistance. A popular rule-of-thumb is that each pane of glass in a conventional glazing system provides R-1 (1 hr·ft^{2}F/BTU) of thermal resistance (e.g., double glazing is R-2).

Emerging technology is creating great potential in the field of glazing system design. Evidence of this progress exists in the wide variety of components (2,3,4)* that have been brought to market or that are currently being researched. Examples of new and technically advanced components include spectrally selective low emissivity (low-e) coatings on glass and on thin plastic films (5,6), solar control coatings (6,7), infrared (IR) transparent glazings (8), anti-reflective surface treatments (9), low conductivity fill gases (10), honeycombs (11,12,13), silica aerogels (14,15), multi-layer film suspension systems, holographic coatings, optically switchable glazings (2,3,4), polarized glazings, and evacuated enclosures. Thermal resistance estimates of windows incorporating some of these advanced components are impressive. An example of one such system is produced by a Swiss manufacturer who claims to have achieved a thermal resistance of close to RSI-1.67 m^2K/W (R-9.5).

---

* References cited in this paragraph are only examples of the many pieces of pertinent literature.

Most windows currently manufactured contain a glazing system that is packaged in the form of a sealed glazing unit (SGU). The SGU typically consists of two panes of glass that are held at a prescribed distance from each other by an edge-seal. The edge-seal isolates the cavity between the glazings - thereby reducing the number of surfaces to be cleaned and creating an enclosure suitable for non-durable coatings and/or substitute fill gases. In contrast to the glazing system, new options to increase the thermal resistance of the SGU edge-seal are only now starting to be pursued. Previously, design improvements had dealt mainly with the requirements of the edge-seal to exclude moisture, provide a desiccant for the sealed space and retain the structural integrity (16) of the SGU. Hence, the thermal bridge created by the edge-seal results in a band at the perimeter of the SGU where the temperature of a glazing can vary significantly as a function of distance from the edge of the glazing. This is an area of increased thermal stress in the glass (17), high energy loss and the site of condensation during cold weather.

It is common for heat transfer through windows to be estimated by treating the frame and glazing areas independently. SGU energy analysis methods have followed a similar course. The SGU can be divided into two areas. The "centre-glass" area is the section of the glazing system that is sufficiently remote from the edge-seal that the heat flux and glazing temperature distributions are uniform and can be characterized as being independent of edge effects. Centre-glass heat transfer is generally simulated as a 1-D phenomenon. The perimeter of the glazing system where the heat transfer is 2-D (or 3-D) and depends upon edge effects (such as the edge-seal conduction or the flow reversal of the fill gas) is customarily called the "edge-glass" section.

Calculation and measurement techniques used prior to 1947 to estimate the thermal resistance of windows were reviewed in detail by Parmalee (18). Parmalee described the guarded hot box, calibrated hot box and hot plate measurement techniques. He presented a large compilation of data and noted that a "considerable range" in measured U-value existed for "approximately similar windows". A calculation procedure was developed but was hindered by a lack of knowledge about natural convection and/or forced convection at the exposed window surfaces and natural convection in the window cavity. Edge effects were neglected. It is interesting to note that the importance of radiant heat exchange with the clear sky was recognized. The emissivity of glass was thought to be 0.937 in contrast to the value of 0.84 that is currently considered valid. Parmalee also pointed out that Peclet believed the thermal resistance of n panes of glass to be $(1+n)/2$ hr·ft^{2}F/BTU. This estimate of thermal resistance differs appreciably from currently accepted values (with the exception of $n=1$).

McCabe and Goss have written an up-to-date review concerning hot-box test methods and calculation procedures (19). This document provides a discussion of the U.S. test standards (ASTM C296, ASTM C976, AAMA 1503.6) plus copies of the Norwegian Standard (NBI-138), the Swedish Standard (SS81 81 29), the Belgian Standard (NBN B62-002) and a working draft of the ASTM Standard being developed (C16:30). A new Belgian Standard for calculating thermal transmission coefficients (U-values) for windows is being developed. This work includes the effect of edge-glass heat transfer and is being prepared in support of the draft ISO Standard which, as recently as May, 1986 (20), did not account for heat loss through edge-seals.

The most recent American Society of Heating, Refrigeration and Air Conditioning Engineers (ASHRAE) estimation procedure, included in the 1989 ASHRAE Handbook of Fundamentals, tabulates centre-glass, edge-glass and frame U-values for the first time as separate quantities. The Canadian Standards Association (CSA) is preparing a standard thermal analysis procedure for windows in support of the Canadian window labeling effort that has been put in motion only recently.

The work described in this document concerns the measurement and computer simulation of centre-glass and edge-glass heat transfer in glazing systems. Some of the research findings have been recorded in the literature. Several undertakings that are not central to this effort (21,22,23,24,25) are described only briefly here in order to provide more detail about the most pertinent research.

Specifically, the topics documented are:

- 1-D centre-glass simulation
- Guarded heater plate measurement of centre-glass heat transfer
- Guarded heater plate measurement of edge-seal thermal resistance
- 2-D computer simulation of natural convection in a vertical, rectangular slot
- 2-D computer simulation (and measurement) of heat transfer in complete sealed glazing units with thermal radiation plus glass and seal conduction included

The description of research on these five topics is contained in Chapters 3 through 7. Each topic contributes toward the end goal of assembling a comprehensive 2-D glazing system analysis. The resulting model can be applied not only for the quantification of heat transfer but also (and perhaps more importantly) for the estimation of temperature profiles and for predicting the presence of condensation.

Each topic presented can be treated as a distinct entity so the various chapters have been written, as much as possible, as stand-alone sources for reference. The nomenclature is explained within the text as it applies in the context of any particular chapter. Care has been taken to leave variable names unchanged in cases where the results and variables of one chapter are reused in a subsequent chapter. Very little overlap exists in the reference material upon which these chapters are based so each chapter has been prepared with its own list of references.

REFERENCES

(1) Selkowitz, S.E., Choi, S.U., DiBartolomeo, D.L., Johnson, R.L., Kessel, J.D., Klems, J.H., Lampert, C.M., Navvab, M., Ne'eman, E., Rubin, M.D., Ryan, J., Wilde, G.M., "Windows and Daylighting", Chapter from FY 1982 Annual Report - Energy Efficient Buildings Program, Lawrence Berkeley Laboratory report W/L-17, 1982.

(2) Lampert, C.M., "Innovative Solar Optical Materials", Optical Engineering, 23(1), pp. 92-97, Jan./Feb., 1984.

(3) Lampert, C.M., "Solar Optical Materials for Innovative Window Design", presented at the Window Energy Show, Las Vegas, NV, August, 1982.

(4) Selkowitz, S.E., Lampert, C.M., Rubin, M., "Advanced Optical and Thermal Technologies for Aperture Control", presented at the Passive and Hybrid Solar Energy Update, Washington, DC, September, 1982.

(5) Selkowitz, S.E., "Transparent Heat Mirrors for Passive Solar Heating Applications", in Passive Systems 78, eds. J. Cook, D. Prowler, 1979, University of Delaware, Newark: American Section International Solar Energy Society, Inc.

(6) "Transparent Heat-Mirror Materials and Deposition Technology", eds. Lampert, C., Selkowitz, S.E., from proceedings of the International Society of Optical Engineering (SPIE) Conference on Optical Coatings for Energy Efficiency and Solar Applications and the Conference on Optical Thin Films, Los Angeles, Jan., 1982.

(7) Valkonen, E., Karlsson, B., Ribbing, C.G., "Solar Optical Properties of Thin Films of Cu, Ag, Au, Cr, Fe, Co, Ni and Al", Solar Energy, Vol. 32, No. 2, pp. 211-222, 1984.

(8) Wright, J.L., "The Computer Simulation of Glazing Systems Which Incorporate Teflon (FEP) Inner Glazings", Proceedings of the ISES/SESCI Intersol'85 Conference, Montreal, Canada (1985).

(9) Rubin, M., Selkowitz, S.E., "Thermal Performance of Windows Having High Solar Transmittance", presented at the American Section of the International Solar Energy Society, Sixth National Passive Solar Conference, Portland, OR, Sept., 1981.

(10) Sorensson, C.B., "Gas - How to Retain It - Gas Filling for Insulating Glass Units", Products Research & Chemical Corporation report, March, 1987.

(11) Stahl, W., Wittwer, V., Pflüger, A., "Transparent Insulation", Solar Energy Materials, 11 (1984) 199-208.

(12) Rommel, M., Wittwer, V., "Flat Plate Collectors for Process Heat with Honeycomb Cover - An Alternative to Vacuum Tube Collectors", ISES Solar World Congress, Hamburg, Sept., 1987.

(13) Hollands, K.G.T., Ford, C., Iynkaran, K., Brundrett, E., "Manufacture, Solar Transmission, and Heat Transfer Characteristics of a Compound-Honeycomb", ISES Solar World Congress, Hamburg, Sept., 1987.

(14) Rubin, M., Lampert, C.M., "Transparent Silican Aerogels for Window Insulation", Solar Energy Materials 7(1983) 393-400.

(15) Hartmann, J., Rubin, M., Arasteh, D., "Thermal and Solar-Optical Properties of Silica Aerogel for Use in Insulated Windows", Proceedings of the 12th Passive Solar Conference, ASES/SESCI, Portland, Oregon, July (1987), pp 42-47.

(16) Barry, C.J., "Air Space Pressures and Sealed Unit Breakage", Ford Glass Ltd., Internal Report, June, 1985.

(17) Solvason, K.R., "Pressures and Stresses in Sealed Double Glazing Units", National Research Council of Canada, Division of Building Research, Technical Paper No. 423., August, 1974.

(18) Parmalee, G.V., "Heat Transmission Through Glass", ASHVE Research Bulletin, Vol. 53, No. 1, July, 1947.

(19) McCabe, M.E., Goss, W.P., "Interim Procedure to Measure the Thermal Performance of Window Systems", prepared by the National Bureau of Standards, Report No. NBSIR 87-3569, for the Bonneville Power Administration, June, 1987.

(20) Proposed ISO Standard, "Thermal Insulation of Glazing: Calculation Rules for Determining the Steady State 'U' Value (Thermal Transmittance) of Double or Mulitple Glazing". ISO TC 160, SC2, Working Group 2, 4th Revision, May, 1986.

(21) Wright, J.L., "Effective U-values and Shading Coefficients of Preheat/Supply Air Glazing Systems," Proceedings of the Solar Energy Society of Canada, Winnipeg, June, (1986) pp. 219-224.

(22) Sullivan, H.F., Wright, J.L., "Recent Improvements and Sensitivity of the VISION Glazing System Thermal Analysis Program," ISES/SESCI 12th Passive Solar Conference, Portland, Oregon, July (1987), pp. 145-149.

(23) Wright, J.L., Sullivan, H.F., "Simulation and Measurement of Windows with Low Emissivity Coatings Used in Conjunction with Teflon Inner Glazings", Proceedings of the ISES Solar World Congress 1987, Hamburg, West Germany, September, (1987) Vol. 1, pp. 625-630.

(24) Wright, J.L., Sullivan, H.F., "Simulation and Measurement of Windows with Metal Films Used in Conjunction with Teflon Inner Glazings". International Conference on Building Energy Management, Lausanne, Switzerland, September (1987) Vol. 4, pp. 5-12.

(25) Baker, J.A., Sullivan, H.F., Wright, J.L., "Screen Graphics Software for the VISION Glazing System Analysis Program," Proceedings of the Solar Energy Society of Canada, Ottawa, June (1988), pp 281-285.

Chapter 2

BACKGROUND

ASHRAE

In North America the American Society of Heating, Refrigerating and Air Conditioning Engineers (ASHRAE) is recognized as an authority on most HVAC matters, and the ASHRAE handbooks (1,2,3,4) as the recommended views on these matters. The source that is now most widely referenced for window information is the Fenestration chapter of the ASHRAE Handbook of Fundamentals. This handbook provides a hand calculation method for estimating the solar gains and heat losses of windows. Data pertaining to a large number of glazing systems and frame types are tabulated in the 1989 ASHRAE Handbook of Fundamentals.

DEFINITION OF WINDOW AREAS

The 1989 ASHRAE Handbook of Fundamentals defines centre-glass, edge-glass and frame areas for the purpose of calculating heat loss. The edge-glass area, A_e, consists of a band of glazing area that is less than 65 mm (2.5 inches) from the sight line. The centre-glass area, A_c, is the remaining sight area of the glazing system. The frame area, A_f, is defined as the projected area of a band of width, w_f, lying outside of the sight line. The value of w_f is a function of frame type and material. Details can be found in Figure 7, Chapter 27 of the 1989 Handbook.

DESIGN CONDITIONS

ASHRAE prescribes two standard conditions for which space cooling and heating loads can be calculated. These conditions, customarily called the standard summer and winter design conditions, represent extreme cases and are intended for use in calculating the peak loads that determine the required size of mechanical equipment. The ASHRAE standard conditions used, almost exclusively, for the simulation of windows are listed in Table 2.1.

Table 2.1: ASHRAE Design Conditions

	Indoor Temperature (°C)	Outdoor Temperature (°C)	Wind Speed (m/sec)	Solar Radiation (W/m^2)
Winter	21	-18	6.7	0
Summer	24	32	3.3	783

The indoor and outdoor temperatures shown in Table 2.1 appear in Figure 17 of the Fenestration chapter in the 1989 ASHRAE Handbook of Fundamentals. Insolation is usually assumed to be beam radiation incident normal to the glazing[!]. Convective heat transfer at the outdoor surface is treated as forced convection and at the indoor surface as natural convection.

U-VALUE

The window U-value is the coefficient of overall heat transfer between the indoor and outdoor environment. The driving potential for this heat transfer is the indoor/outdoor (dry bulb) air temperature difference, as shown in Equation 2.1.

$$q_{hl} = U \cdot (T_{in} - T_{out}) \qquad (\mathrm{W/m^2 \ or \ BTU/hr \cdot ft^2}) \qquad (2.1)$$

where, q_{hl} = indoor to outdoor heat flux (or heat loss)

T_{in}, T_{out} = indoor and outdoor air temperatures, respectively

In reality, the heat transfer is driven not only by the local air temperatures but also by the mean radiant temperatures to which the window is exposed. The mean radiant temperature of the indoor environment is generally assumed to be equal to the indoor air temperature. This is a safe assumption in the usual case where the

---

[!] The presence of beam solar radiation in the summer design condition contradicts the suggested ASHRAE procedure for calculation of thermal radiation exchange on the outdoor side which calls for the assumption of a fully cloudy sky.

exchange of thermal radiation takes place between a small window and a large room with indoor surfaces at (or near) the room air temperature. This is clearly not valid in rooms where the window is exposed to other large areas of glazing surface (e.g., greenhouse, atrium) or other cooled or heated surfaces.

The mean radiant temperature of the outdoor environment is frequently assumed to be equal to the outdoor air temperature. This assumption may be badly in error. A window heat loss calculated using a tabulated U-value and based on outside air temperature will be in error if the window is exposed to areas of clear sky — more so for a window with high U-value such as single glazing.

The U-value includes the effect of thermal resistances between each side of the window and the environment. The choice of appropriate indoor and outdoor thermal resistances has been the topic of a long-standing debate. The convective and radiative modes are often combined to form the indoor and outdoor conductances, h_{in} and h_{out}, which are usually called "film coefficients". Typical values of h_{in} and h_{out} are 8 and 23 W/m^2C, respectively. It is impossible to compare the merit of window designs based on U-values from different sources unless the corresponding film coefficients are known. It is a simple matter to determine h_{in} and h_{out} in calculated results but the film coefficients for measured results will differ depending on the particular hot-box used and the specific test method and test condition.

SHADING COEFFICIENT

The Shading Coefficient (SC) is a measure of the solar gain (including both direct solar gains and solar gains absorbed at the glazing system and redirected to the conditioned space) that a glazing system will deliver. SC is the ratio of solar heat gain through a glazing system to the solar heat gain through a reference glazing under the same set of conditions. The reference glazing is 3 mm double-strength (DSA) sheet glass. DSA is soda-lime glass or common window glass and is frequently called clear glass. If the ratio of solar gain to total incident solar radiation is denoted F, then:

$$SC = \frac{F(\text{glazing system})}{F(DSA)} \qquad (2.2)$$

The value of F for the reference glazing under the standard summer condition is $F(DSA)=0.87$.

ASHRAE tabulates values of solar heat gains for the reference glazing and calls them solar heat gain factors, SHGF. Knowing SHGF for a particular location, orientation, time, etc. it is possible to estimate the solar gain for any glazing system, q_{sg}, as:

$$q_{sg} = SC \cdot SHGF \qquad (W/m^2 \text{ or } BTU/hr \cdot ft^2) \qquad (2.3)$$

The value of SC, by definition, for the reference glazing, DSA, is $SC=1$. Glazing systems will generally have $SC<1$ with multiple glazings, tinted glass, solar reflective coatings and low-e coatings being examples of components that reduce solar gains. Single glazings that reflect less or absorb less solar radiation than the DSA reference glazing (e.g., low-iron glass, borosilicate glass, clear glass less than 3 mm thick) will have $SC>1$.

NET ENERGY GAIN OR LOSS

The 1989 ASHRAE Handbook presents a hand method for calculating the centre-glass U-value of a glazing system. This method is cumbersome for multipane systems and is not appropriate for glazing systems with glazings that are not opaque to thermal radiation. The edge-glass U-value, U_e, can be estimated as a function of centre-glass U-value, U_c, and edge-seal construction. U-values for a variety of generic frame designs, U_f, are given. The U-value for the total window area, U_w, is then calculated by weighting centre-glass, edge-glass and frame U-values according to their corresponding areas.

$$U_w(A_c+A_e+A_f) = U_cA_c + U_eA_e + U_fA_f \qquad (2.4)$$

The rate of net heat gain experienced by the window, Q_{net}, can be found by taking the difference between the solar gain (see Equation 2.2) applied over the aperture area and the heat loss (using the total window U-value in Equation 2.1) applied over the total window area.

$$Q_{net} = SC \cdot SHGF \cdot (A_c + A_e) - U_w (T_{in} - T_{out}) \cdot (A_c + A_e + A_f) \quad \text{(W or BTU/hr)} \qquad (2.5)$$

CURRENT GLAZING SYSTEM DESIGN PRACTICE

Knowing that the majority of heat transfer between the glazings in a conventional double glazed (CDG) system takes place by the exchange of thermal radiation, it can be seen that the thermal resistance will increase dramatically if the emissivity of one of the interpane surfaces is reduced. Low-e coatings have been developed specifically for this application and their use is becoming commonplace.

Two types of low-e coatings are available. The type most frequently encountered is a vacuum sputtered "stack" where the components of the stack are sub-wavelength layers, typically in a dielectric/metal/dielectric arrangement. The metal layer provides the high reflectance to longer (thermal) wavelengths and the dielectic layers reduce reflection in the visible wavelength band. The sputtered low-e coatings are called "soft low-e" because they can be damaged if handled or exposed to moisture. They must be sealed in an SGU for protection. The second type of coating is known as pyrolitic low-e because it is a material that is "fired" onto glass during the float-line process. This type of coating is durable and is sometimes called hard low-e.

Metal coatings sputtered onto glass for the purpose of reflecting solar radiation in commercial applications also have low emissivities because of the inherent reflective nature of the metal surface.

The design of low-e coatings has been successful in that they offer reduced surface emissivity while remaining virtually undetectable to the eye. Soft low-e offers emissivity near or below $\varepsilon_{le}=0.1$. Higher emissivity values, typically ε_{le}=0.35 to 0.45, are available with hard low-e. A double glazed system with a soft low-e coating (DGLE) has 50% more thermal resistance than CDG or about the same thermal resistance as conventional triple glazing (CTG) (i.e., R-3 vs. R-2 for CDG). The addition of a low-e coating to clear glass more than doubles the portion of incident solar radiation absorbed by the glazing. Therefore, low-e coatings can substantially reduce the shading coefficient of a glazing system. In cold climates where solar gains are desirable the low-e coating, if one is to be used, is placed on the indoor glazing rather than the outdoor glazing so that the majority of the solar radiation absorbed at the low-e coating is redirected to the indoor space. Overheating can be severe if a low-e coating is located on one of the intermediate panes in a multi-pane glazing system having high thermal resistance because the energy from solar radiation absorbed at the coating cannot readily escape.

Once a low-e coating has been incorporated between two glazings the dominant mode of heat transfer becomes conduction/convection. In order to further increase thermal resistance many manufacturers are now putting a low conductivity substitute fill gas between the SGU panes. Argon is the fill gas used almost exclusively in North America. Its thermal conductivity is about 2/3 as great as the conductivity of air. In Europe, sulphur hexafluoride (SF_6) is sometimes used, either by itself or mixed with argon, because of its ability to attenuate noise. The thermal conductivity of SF_6 is about 1/2 the conductivity of air.

A double glazed SGU with a soft low-e coating and argon (DGLEA) will have centre-glass thermal resistance in excess of R-4. More exotic glazing system designs with more glazings, coatings with even lower emissivity (ε_{le}<0.05 is now possible) and krypton fill-gas (conductivity ≈ 1/3 the conductivity of air) offer thermal resistance in the R-6 to R-10 range if the glazing units are sealed. Thermal resistance beyond R-10 is available if a method of pressure equalization is used so that the thickness of the sealed space can safely exceed the current limit of about 25 mm.

REFERENCES

(1) ASHRAE. 1990. ASHRAE Handbook – 1990 Refrigeration, Atlanta: American Society of Heating, Refrigerating and Air Conditioning Engineers, Inc.

(2) ASHRAE. 1989. ASHRAE Handbook – 1989 Fundamentals, Atlanta: American Society of Heating, Refrigerating and Air Conditioning Engineers, Inc.

(3) ASHRAE. 1988. ASHRAE Handbook – 1988 Equipment, Atlanta: American Society of Heating, Refrigerating and Air Conditioning Engineers, Inc.

(4) ASHRAE. 1987. ASHRAE Handbook – 1987 Systems and Applications, Atlanta: American Society of Heating, Refrigerating and Air Conditioning Engineers, Inc.

Chapter 3

GLAZING SYSTEM ANALYSIS

VISION 1-D GLAZING SYSTEM ANALYSIS

In light of the increasingly complex nature of glazing system design, as well as the fundamental differences in infrared (IR) properties of some of the plastic films now available, it has become apparent that conventional calculation methods (such as the method provided by ASHRAE) or "rule of thumb" prediction methods are no longer adequate. In order to fill this void, and to support the research effort at the National Research Council of Canada, a glazing system computer simulation program called VISION was written (1-6). VISION performs a 1-D, centre-glass, two-band (solar and thermal wavelengths) analysis. The algorithm used in VISION is based on the method presented in references (7) and (8). A program called Window (based on the work of Rubin(9)) is maintained by the Lawrence Berkeley Laboratory (University of California) and incorporates models almost identical to those found in VISION.

VISION has been used since 1982 as the basis of a variety of studies including the development of a simplified seasonal thermal performance calculation method (10,11), the illustration of basic heat transfer characteristics unique to a specific class of glazing system (12,13,14), the comparison of a selection of commercially available and prototype glazing systems (15,16) plus a variety of other modeling studies (17,18,19). Capabilities of VISION that provide a significant improvement over previous methods include the ability to model multiple glazings, fully or partially IR transparent glazings, sloped glazing and substitute fill gases. Inter-pane convective heat transfer is handled using the correlations of ElSherbiny, Raithby and Hollands (20). The more important VISION capabilities are listed in Table 3.1.

The VISION software has been undergoing development at the University of Waterloo as part of the current research effort. Recent modifications include a revised method for calculating U-value and shading coefficient, user specified design conditions (while accounting for clear sky conditions if necessary), an input library for optical properties of various glazings, additional fill gas mixtures and graphic output. Reference (2) is the most up to date publication that documents all of the VISION models in full detail.

Table 3.1: Capabilities of the VISION Glazing System Analysis Program

- Up to 6 glazings
- Tilt from 0 to 180°
- Substitute fill gases and gas mixtures
- Partial or hard vacuum
- IR transparent glazings
- Asymmetric optical properties for any glazing
- Input library with optical properties of individual glazings
- ASHRAE winter, ASHRAE summer and user-specified design conditions including capability for clear-sky analysis
- Optional hour-by-hour analysis with side fin, overhang and setback shading at any tilt
- Plotted, screen or printed graphic output

VISION was designed to calculate centre-glass U-value and SC for the ASHRAE winter and summer design conditions. A completed analysis provides a solution for the temperature at each glazing, T_i, and the heat flux between glazing pairs, q_i, in a multi-glazing array exposed to a particular set of conditions. Figure 3.1 shows the location of temperatures and heat flux quantities as well as other details regarding the nomenclature.

Under the winter design condition no solar radiation is incident. Therefore, it is not possible to calculate a shading coefficient and the U-value is found simply by dividing the heat flux from the building interior to the indoor glazing, $-q_1$, by the indoor/outdoor temperature difference.

Under the summer condition, solar radiation is present and the calculation is more complex. VISION, in its original configuration, could treat this situation by modeling a glazing system under two separate conditions. First, the summer condition was simulated but the solar radiation was set to zero. In this case the U-value was calculated in the same manner as it was for the winter condition. Second, the glazing system was modeled with solar radiation present but with the indoor/outdoor temperature difference set to zero. In this case the heat flux due to the indoor/outdoor temperature potential was taken as zero and the solar gain was equated to the sum of the directly transmitted solar radiation plus the total heat flux from the indoor glazing to the building interior, q_1. Consequently, this solar gain could be used to calculate shading coefficient.

Recently, VISION was modified to calculate both U-value and SC by simulating a single environmental condition involving indoor/outdoor temperature difference and incident solar radiation simultaneously. In order to carry out this calculation it is necessary to evaluate the heat flux from the indoor glazing to the building interior, q_1, as two parts - one driven by indoor/outdoor temperature difference and the other resulting from the presence of solar radiation absorbed at each of the various glazings. The amount of solar radiation that is absorbed at the i^{th} glazing, S_i, is calculated using the embedding technique of Edwards (21). The portion of S_i that eventually finds its way to the building interior, N_i, is equal to the portion of the total thermal resistance that exists between the i^{th} glazing and the outdoor environment.

$$N_i = \sum_{j=i}^{n-1} R_j / R_{tot} \qquad (3.1)$$

where the total thermal resistance is:

$$R_{tot} = \sum_{j=1}^{n-1} R_j \tag{3.2}$$

and the thermal resistance between node j and j+1 is:

$$R_j = (T_{j+1} - T_j)/q_j \tag{3.3}$$

The total flux of incident solar radiation that makes its way to the building interior, q_{sol}, is:

$$q_{sol} = \sum_{j=1}^{n-1} N_j S_j \tag{3.4}$$

Finally, the shading coefficient is found using:

$$SC = q_{sol} / q_{ref} \tag{3.5}$$

where q_{ref} is equal to q_{sol} calculated for the reference DSA glazing under the same environmental conditions. The centre-glass U-value is given by:

$$U_c = 1 / R_{tot} \tag{3.6}$$

VISION provides the opportunity of modeling user specified conditions. Therefore, it is possible to calculate the summer U-value or shading coefficient according to the older method by specifying the ASHRAE summer condition while first setting the solar radiation then the indoor/outdoor temperature difference to zero.

When describing a glazing system it is possible to draw optical data from a library of glazing data files. This library resides in the computer storage. If data for a particular glazing do not exist in the library it is possible to enter the information manually and have it recorded.

The VISION output file provides a detailed description of the glazing system as well as a variety of simulation results including centre-glass U-value and SC. The user can generate tables of total window U-value estimated using the ASHRAE method. It is possible to produce graphic output on an HP-7470A or HP-7475A plotter. An example of plotted output for an argon filled glazing system is shown in Figure 3.2. The temperature profile is shown superimposed on the glazing system cross-section. Incident solar radiation and the various amounts of absorbed solar radiation, S_i, are shown above the glazing profile. The heat flux quantities, q_i, their components and their flow directions are shown below the glazing profile.

In general, VISION results have been found to be in good agreement with the available published data (5). Calculated U-values are typically within 10% of the published data for commercial windows. Discrepancies are smaller in cases where input variables (e.g., surface emissivities, pane spacings, etc.) of specific test samples can be measured. The discrepancy between calculated and published shading coefficients is consistently less than 3%.

EDGE-GLASS ANALYSIS

The models incorporated in VISION provide a reliable means of undertaking the centre-glass thermal analysis of glazing systems but the results from these 1-D models cannot offer information about edge-glass performance. More than one 2-D method exists for estimating edge-glass heat loss (22,23). These methods do not account for the mechanisms of edge-glass heat transfer in full detail. They provide a 2-D treatment of conduction but they do not model the motion of the fill gas or radiative exchange in the SGU cavity as 2-D phenomena. These simplifications make it impossible for the analysis to distinguish between the top and bottom edges of the SGU. However, the existing edge-glass models are useful tools for calculating total heat loss. Their accuracy is not as critical as the accuracy of the centre-glass model because it is known that edge-glass heat loss typically constitutes 10% or less of the total window heat loss (22,23).

Since the current edge-glass analysis methods do not provide a complete 2-D treatment it can be seen that there is a need for a detailed 2-D model if realistic predictions of SGU bottom-edge heat loss and glazing temperature profiles are to be obtained. It will be possible to explore the implications of the various fill gases, optical coatings, glazing materials and new edge-seal designs with respect to condensation resistance only after the full 2-D model is in place.

1-D PREHEAT/SUPPLY AIR GLAZING SYSTEM MODEL

A study involving the modification of the VISION code was undertaken in order to model various preheat/supply air glazing system configurations. In this type of window, ventilation air is brought into the building between the panes of the glazing system and the energy recovered by the preheating of the air flow is credited to the thermal performance of the window itself. Thus, the net energy flow between the conditioned space and the window was used to calculate the "effective" U-value and "effective" shading coefficient. The use of effective window performance parameters permits the quantification of energy flows to or from the supply air glazing system (in the same way that conventional glazing systems are treated) without the necessity of modeling the detailed mechanisms of energy transport within the glazing system itself.

The conductive/convective heat transfer rates from the glazings adjacent the forced air flow were determined using a two dimensional analytical solution of basic equations describing the elemental energy balance in the fluid flow (24). Eleven preheat/supply air glazing configurations were simulated with designs ranging from a simple double glazed system to more complex configurations involving the use of up to four glazings, low-e coatings and argon fill gas. Interactions between supply air flow location, the number of panes and low-e coating placement were examined with indoor pane temperature and supply air delivery temperature calculated in each case. Effective thermal resistance values of preheat windows were found to be appreciably larger than those of conventional windows. Details of this study are available in reference (25), a copy of which is included in Appendix A.

REFERENCES

(1) Ferguson, J.E., Wright, J.L., "A Computer Program to Evaluate the Thermal Performance of Super Windows", National Research Council of Canada, Division of Energy report no. Passive-10, June, 1984.

(2) Wright, J.L., Sullivan, H.F., "VISION: A Computer Program to Evaluate the Thermal Performance of Innovative Glazing Systems - Reference Manual", report produced by the Advanced Glazing Systems Laboratory, Department of Mechanical Engineering, University of Waterloo, Waterloo, Canada, 1987.

(3) Ferguson, J.E., Wright, J.L., "VISION: A Computer Program to Evaluate the Thermal Performance of Super Windows", Proceedings of the 1984 SESCI Conference, Calgary, Canada.

(4) Sullivan, H.F., Wright, J.L., "Recent Improvements and Sensitivity of the VISION Glazing System Thermal Analysis Program", Proceedings of the 12th Passive Solar Conference, ASES/SESCI, Portland, Oregon, July (1987), pp 145-149.

(5) Wright, J.L., Sullivan, H.F., "VISION: A Computer Program for the Detailed Simulation of the Thermal Performance of Innovative Glazing Systems", International Conference on Building Energy Management, Lausanne, Switzerland, September (1987), Vol.2, pp 179-185.

(6) Baker, J.A., Sullivan, H.F., Wright, J.L., "Screen Graphics Software for the VISION Glazing System Analysis Program," Proceedings of the Solar Energy Society of Canada, Ottawa, June (1988), pp 281-285.

(7) Wright, J.L., "Free Convection in Inclined Air Layers Constrained by a V-Corrugated Teflon Film", M.A.Sc. thesis, Department of Mechanical Engineering, University of Waterloo, Waterloo, Canada, 1980.

(8) Hollands, K.G.T., Wright, J.L., "Heat Loss Coefficients and Effective $\tau\alpha$ Products for Flat-Plate Collectors with Diathermanous Covers", Solar Energy, Vol. 30, pp. 211-216, 1982.

(9) Rubin, M., "Calculating Heat Transfer Through Windows", Energy Research, Vol 6, pp. 341-349, 1982.

(10) Harrison, S.J., Barakat, S.A., "A Method for Comparing the Thermal Performance of Windows", ASHRAE Transactions, vol. 89., part 1, pp 3-11, 1983.

(11) Barakat, S.A., "Comparison of the Thermal Performance of Glazing Systems", Proceedings of the ISES/SESCI Intersol'85 Conference, Montreal, Canada, 1985.

(12) Wright, J.L., "The Computer Simulation of Super Window Glazing Systems which Incorporate Teflon (FEP) Inner Glazings," Proceedings of the ISES/ SESCI Intersol'85 Conference, Montreal, Canada, 1985, Vol. 1, pp. 267-271.

(13) Wright, J.L., Sullivan, H.F., "Simulation and Measurement of Windows with Metal Films Used in Conjunction with Teflon Inner Glazings," International Conference on Building Energy Management, Lausanne, Switzerland, September (1987), Vol. 4, pp. 5-12.

(14) Wright, J.L., Sullivan, H.F., "Simulation and Measurement of Windows with Low Emissivity Coatings Used in Conjunction with Teflon Inner Glazings," ISES Solar World Congress, Hamburg, West Germany, September (1987), Vol. 4, pp. 3136-3140.

(15) Ferguson, J.E., Wright, J.L., "Optimization of Glazing Design for Residential Use", Proceedings of the ASHRAE/DOE/BTECC, Thermal Performance of the Exterior Envelope of Buildings Conference, Clearwater, Florida, U.S.A., December, (1985).

(16) Ferguson, J.E., Wright, J.L., "VISION: A Computer Program to Evaluate the Thermal Performance of Super Windows," Proceedings of the Solar Energy Society of Canada, page 241, Calgary, 1984.

(17) Baker, J.A., Sullivan, H.F., Wright, J.L., "A Study of Pane Spacing in Glazing Systems," Proceedings of the Solar Energy Society of Canada, Penticton, June (1989).

(18) Baker, J.A., Sullivan, H.F., Wright, J.L., "Window (Glazing and Frame) Heat Transfer Modelling," 1990 ASHRAE Winter Meeting, Atlanta, Georgia, February 11-14, ASHRAE Transactions, Vol. 96, Pt. 1, (1990).

(19) Sullivan, H.F., Wright, J.L., "Recent Improvements and Sensitivity of the VISION Glazing System Thermal Analysis Program," ISES/SESCI 12th Passive Solar Conference, Portland, Oregon, July (1987), pp. 145-149.

(20) ElSherbiny, S.M., Raithby, G.D., Hollands, K.G.T., "Heat Transfer by Natural Convection Across Vertical and Inclined Air Layers", Journal of Heat Transfer, Vol. 104, Feb., 1982.

(21) Edwards, D.K., Solar absorption by each element in an absorber-coverglass array, Technical Note, Solar Energy, Vol. 19, pp. 401-402, 1977.

(22) Klems, J.H., Riley, S., "Window Nighttime U-values: A Comparison Between computer Calculations and MoWiTT Measurements," presented at the 1990 ASHRAE Winter meeting, Atlanta, preprint no. AT-90-7-5.

(23) Enermodal Engineering Ltd., "The Effect of Frame Design on Window Heat Loss, Phase 1," report prepared for the Renewable Energy Branch of Energy, Mines and Resources, Ottawa, 1987.

(24) Cess, R.D., Shaffer, E.C., "Summary of Laminar Heat Transfer Between Parallel Plates with Unsymmetrical Wall Temperatures", Journal of the Aero/Space Sciences, pp. 538, August, 1959.

(25) Wright, J.L., "Effective U-values and Shading Coefficients of Preheat/Supply Air Glazing Systems," Proceedings of the Solar Energy Society of Canada, Winnipeg, June, (1986) pp. 219-224.

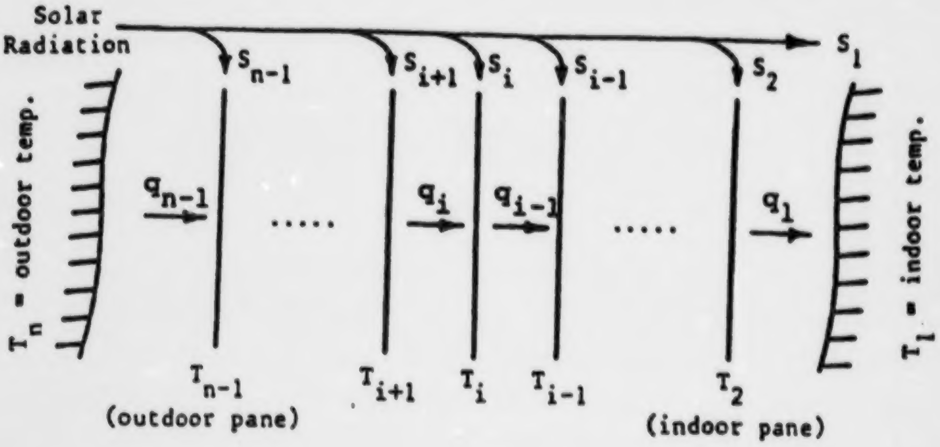

Solar Radiation
S_{n-1}
S_{i+1}
S_i
S_{i-1}
S_2
S_1
T_n = outdoor temp.
q_{n-1}
.....
q_i
q_{i-1}
.....
q_1
T_1 = indoor temp.
T_{n-1}
(outdoor pane)
T_{i+1}
T_i
T_{i-1}
T_2
(indoor pane)

Figure 3.1: 1-D Glazing System Thermal Analysis Geometry

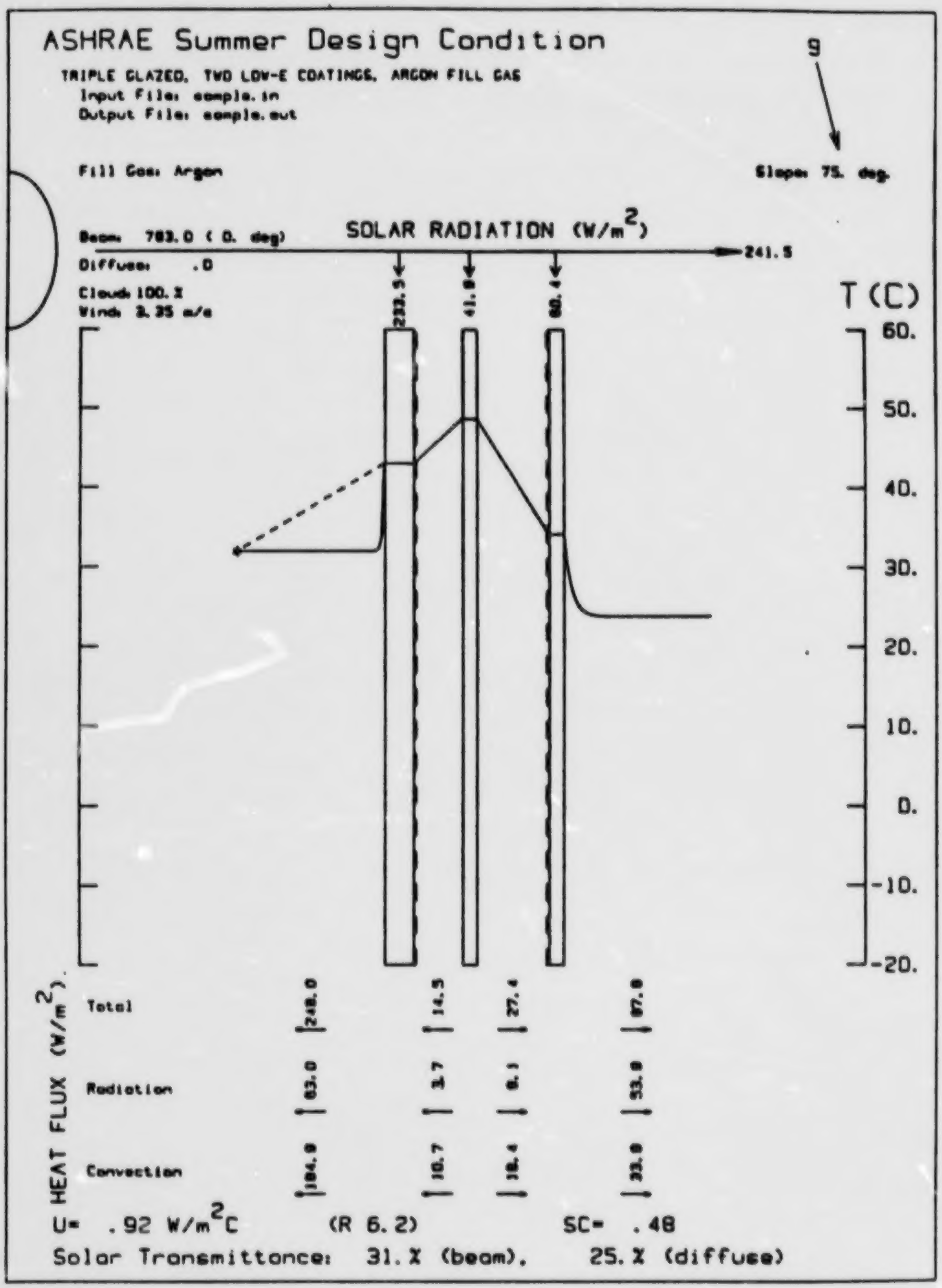


Figure 3.2: Sample Graphic Output from VISION

Chapter 4

CENTRE-GLASS U-VALUE MEASUREMENT USING THE UNIVERSITY OF WATERLOO NATURAL CONVECTION APPARATUS

INTRODUCTION

Many precise heat transfer measurements have been carried out using the University of Waterloo Natural Convection Apparatus (UWNCA). This apparatus has been adapted and used during this study to perform centre-glass U-value measurements on a variety of prototype glazing systems.

The UWNCA consists primarily of two parallel copper plates (each 635x635x12.7 mm (25x25x0.5 in)), positioned facing each other, that can be maintained at different but constant temperatures. Each plate is held at its desired temperature by a constant temperature circulating bath that pumps a steady flow of water and glycol through a manifold of tubes attached to the back of the copper plate. The warm copper plate contains three guarded heater plates. These three heater plates (each 200x200 mm (7.875x7.875 in)) lie in recesses located in a row across the centre of the warm copper plate. This heater plate arrangement is shown in Figure 4.1. It is possible to measure the heat transfer that occurs over the face of each of the guarded heater plates. An earlier version of the copper plates contained only one guarded heater plate at the centre of the warm copper plate. This configuration was capable of measuring heat transfer at the centre region only and no heat transfer information at the edges could be gathered.

Each heater plate is positioned in a recess such that the flat face of the heater plate is flush with the surface of the warm copper plate as indicated in Figure 4.2. Each heater plate is also made of copper. Thus, the full surface of the warm copper plate appears to be flat and continuous. The experimental measurement of heat transfer involves the adjustment of the electrical power supplied to the nichrome wire in a heater plate until zero temperature difference is measured between that heater plate and the larger copper plate in which it is embedded. Once this condition is reached there can be no heat transfer between the heater plate and the warm copper plate and all of the electrical power supplied is transferred to the cold copper plate. The temperature difference between the heater plate and the warm copper plate is sensed by a heat flux meter that is sandwiched between these two plates. The heat flux meter is a flexible, rubbery disk that contains a thermopile with a large number of junctions. Since the output voltage of the heat flux meter requires no electronic conditioning, no danger of a voltage offset exists. Therefore, it is an ideal device for detecting the desired null condition.

Many basic research studies have been carried out using the UWNCA all of which quantified heat transfer due to natural convection that occurred in the gap between the copper plates. These studies have uncovered information about the stability of horizontal and inclined air layers heated from below (1,2), heat transfer in vertical and inclined air layers (2,3,4,5,6,7,8), heat transfer in air layers where one surface is v-corrugated (9) and heat transfer across air layers containing a wide variety of convection suppression devices (1,10,11,12,13,14,15). In each case, a Nusselt number (Nu), a dimensionless representation of the rate of heat transfer due to convection, was defined and examined as a function of those variables that drive and determine the characteristics of the convective flow. In configurations where the radiative and convective heat transfer modes are not coupled (e.g., by a convection suppression device such as a honeycomb) and where the air layer geometry has been fixed (e.g., cavity aspect ratio, tilt) Nu is found to be solely a function of the Rayleigh number, Ra. Treating air as an ideal gas, Ra can be expressed as:

$$Ra = \frac{P^2\, g\, \Delta T\, C_p\, L^3}{R^2\, k\, Z^2\, \mu\, T_m^3} \qquad \text{,dimensionless} \qquad (4.1)$$

where, P = pressure (absolute)
g = gravitational acceleration
ΔT = temperature difference across the air layer
C_p = specific heat of air
L = thickness of air layer (plate spacing)
R = universal gas constant
k = thermal conductivity of air
Z = compressibility factor for air
μ = viscosity of air
T_m = mean (absolute) temperature of air layer (calculated as the arithmetic mean of the adjacent surfaces temperatures)

It can be seen that, aside from the properties of the air itself (C_p, k, μ, Z) and the constants (R, g), Ra is a function of the conditions imposed on the air layer, namely T_m, L, ΔT, and P. During experimental convection studies many heat transfer measurements were made for imposed conditions that represented a wide range of Ra. The value of Ra was held constant during each measurement. Ra was varied between measurements but was not controlled by changing T_m, L or ΔT. Rather, the UWNCA plates were housed in a pressure vessel providing the unique capability of controlling Ra by controlling the pressure of the air layer between the plates. Pressure control, control of electrical current supplied to the heater plates and data acquisition was carried out by an automated system in order to gather the large sets of data necessary for natural convection studies.

In contrast, the measurement of glazing system U-value using the UWNCA plates is a relatively simple procedure. U-value testing is carried out at atmospheric pressure so the measurement does not need to be done inside a pressure vessel. Comparatively few measurements are needed and adjustments made in the course of the experimental procedure can be performed manually.

GLAZING SYSTEM HEAT TRANSFER MEASUREMENT

Adaptation of Apparatus

Each glazing system tested is placed between the two flat copper plates. It is desirable that the glazing system under test be the same size as the copper plates. In order to prevent direct contact between the copper and glass surfaces, and to promote good thermal contact, sheets of neoprene (rubber) are placed between the glazing system and the copper plates. Figure 4.3 shows a cross-section view of a triple glazed SGU installed between the plates.

Once the glazing system is securely fastened in place the plates are fixed at the desired tilt angle. Next, the outside of the copper plates/glazing system assembly is completely covered with a thick layer of fibreglass insulation in order to minimize thermal losses from (or gains to) what would otherwise be the exposed backsides of the copper plates. The insulation also prevents the formation of condensed water vapour and/or frost on the cold copper plate and headers.

After the insulation is in place the constant temperature baths that feed the two copper plates can be started. The time required for the temperatures of these baths to reach steady state depends on the thermostat settings and on the capacity for heating or cooling that the baths can provide. Generally, the warm side temperature is near room temperature and the warm bath reaches steady state within ten or fifteen minutes. The cold side bath takes several hours to reach a temperature of -10 °C (14 °F).

Measurement Procedure

In order to measure the heat flux from the exposed surface of a guarded heater plate it is necessary to measure the rate at which electrical energy is supplied to the heater plate, Q_e, and the heat flux meter output voltage, HFMV. Electrical current is supplied to the heater plate by a regulated D.C. power supply and the rate at which energy is dissipated is simply:

$$Q_e = I_h \cdot V_h \tag{4.2}$$

where, I_h = current measured through the heater wire
V_h = voltage drop through the heater wire

The heater wire voltage, V_h, is measured directly using a digital volt meter. During previous convection studies the heater current, I_h, was determined by measuring the voltage across a calibrated shunt resistance. More recent studies have included the accurate calibration of the heater wire resistance (nominally 10.45Ω) which allows I_h to be determined directly from V_h (and the temperature of the heater plate). Currently, when experiments are performed to determine glazing system U-values, I_h is measured directly using a digital multimeter.

The measurement of heat transfer during studies of natural convection has involved the adjustment of Q_e while observing the resulting change in HFMV. This adjustment was carried out until HFMV was small and the difference in temperature between the heater plate and the warm copper plate was also small (8). It is necessary that HFMV be small enough that the boundary condition along the warm side of the convecting air layer is close to being isothermal so that no irregularities in the resulting flow or temperature fields are created. The heat flux from the face of the heater plate, q, is then calculated as:

$$q = (Q_e - \alpha \cdot \mathrm{HFMV})/A_{hp} \tag{4.3}$$

where, α = heat flux meter calibration constant
A_{hp} = face area of the heater plate

This method of determining the heater plate heat flux was appropriate in that it was not necessary to adjust the power supply setting to the point where HFMV was exactly zero. Consequently, a large number of measurements could be made with a minimum of time and effort but it was necessary to perform a detailed calibration experiment prior to the formal experimentation in order to determine α.

When a glazing system heat transfer measurement is made it is more convenient to adjust the power supply setting until HFMV is zero. In this case the heat flux at the face of the heater plate can be calculated according to Equation 4.3 and no calibration constant is needed ($q = Q_c/A_{hp}$).

During the experiment the power supply is adjusted (for each heater plate) a number of times before the heat flux meters are satisfactorily balanced. After each adjustment it is necessary to wait for at least twenty minutes before steady state is re-established and new HFMV readings can be taken. In order to monitor the time dependent changes in the heat flux meter signal, HFMV is recorded using a strip chart recorder. A sample of one of these records is shown in Figure 4.4. It can be seen in Figure 4.4 where HFMV reached steady state and a reading was made on three occasions. The power supply was adjusted following the first two readings. It is also noteworthy to see in Figure 4.4 that HFMV has a consistent cyclic swing of approximately ±0.2 mV. This fluctuation is caused by the on/off nature of the heaters in the constant temperature baths as their thermostats trip on and off. Even though the constant temperature baths are capable of temperature regulation to within a very fine tolerance (The manufacturer claims regulation to ±0.01°C.), the heat flux meters are sensitive enough that they can detect the operation of the bath heaters. The reading of the HFMV value that is done "by eye" using the HFMV chart recorder trace is the largest source of uncertainty in the measurement. An estimated accuracy in HFMV of ±0.1 mV corresponds to a possible error in measured heat flux of approximately ±0.56 W/m^2 (depending on the particular test section being used) which translates into a U-value error of less than ±1% for a glazing system with an indoor/outdoor U-value of 2.0 W/m^2K (R-value = 2.8 $hr \cdot ft^2 \cdot {}^\circ F/BTU$) tested at a temperature difference of 20°C. This relative level of uncertainty would double if the level of heat flux were halved by testing at a lower temperature difference or by testing a glazing system with a higher thermal resistance. Errors associated with the measurement of temperature difference between the plates and the rate of energy dissipation at the heater plates are much smaller.

As the experiment proceeds and the various readings are compiled it is useful to graph the measured values of Q_e as a function of the measured values of HFMV. Figure 4.5 shows a sample set of results for one experimental run. The heat flux meter voltage readings listed in Figure 4.5 for the bottom heater plate ($HFMV_3$) were taken from the chart recorder trace of Figure 4.4. Generally, the plotted points lie in a straight line and can be used to determine the next setting of the power supply voltage. If the heater plate is eventually balanced such that HFMV is very close to being zero but not exactly zero it is possible to estimate the desired value of Q_e (at HFMV=0) graphically by constructing a straight line using the previously measured points. In fact, this is equivalent to using Equation 4.3 where the slope of the line through the points gives α, the heat flux meter calibration constant that is appropriate for that particular test section.

Once the heat flux, q, has been measured, the glass-to-glass thermal conductance of the glazing system, C, can be calculated. This is done using Equation 4.4.

$$C = [(\Delta T_{pp}/q) - 2R_n]^{-1} \tag{4.4}$$

where, ΔT_{pp} = temperature difference between the warm and cold copper plates

R_n = thermal resistance of one neoprene sheet

The value of R_n was measured directly using the UWNCA plates.

The temperature difference between the copper plates is measured using a copper/constantan thermopile with six thermocouple junctions in each plate. A curve fit based on information found in reference (16) is used to convert the thermopile output voltage, EMF, to temperature difference. This equation, valid from 323 K to 423 K, is:

$$\begin{aligned}\Delta T_{pp} = {} & 9.259 \times 10^{-2} \cdot \mathrm{EMF/DEN} \\ & + 2.049 \times 10^{-7} \cdot \mathrm{EMF}^2/\mathrm{DEN}^3 \\ & + 9.072 \times 10^{-13} \cdot \mathrm{EMF}^3/\mathrm{DEN}^5 \quad \text{,Celsius degrees} \end{aligned} \tag{4.5}$$

where, $\mathrm{DEN} = 2.14 \times 10^{-2} + 4.781 \times 10^{-5} \cdot T_h$

T_h = temperature of the warm copper plate, °C

Note also that the temperature difference between any of the heater plates and the cold copper plate can be slightly different than ΔT_{pp} when HFMV is not equal to zero but no correction is needed when HFMV=0.

The measured C-values can be converted to the more familiar U-value form by using suitable indoor and outdoor conductances (h_{in} and h_{out}, respectively) to modify the measured C-value.

$$U = \left(C^{-1} + h_{in}^{-1} + h_{out}^{-1} \right)^{-1} \tag{4.6}$$

Since the procedure does not involve the measurement of heat transfer between the glazing system and its environment, many of the problems involving the validity and experimental reproducibility of h_{in} and h_{out} are eliminated.

MEASUREMENT RESULTS

Earlier work regarding solar collectors has illustrated that highly infrared (IR) transparent plastic films are well suited for use as intermediate covers when used adjacent to a low emissivity absorber surface (17,18). The current research extends this work. Two experimental studies were undertaken using the UWNCA in order to measure the centre-glass C-values of a variety of glazing systems whose design was based on the same concept (19). One study involved the use of metal solar-control coatings and the other included glazing systems with low-e coatings. In both cases 0.0254 mm (0.001 inch) thick fluorinated ethylene-propylene (FEP) film* was used to construct intermediate glazings. FEP film was chosen for study because of its excellent chemical and UV stability, very high solar and visible transmittance, high IR transmittance and off-the-shelf availability.

In the first of these two studies (20) six prototype glazing systems with metal solar-control coatings were tested. All of these glazing systems consisted of two 6 mm (0.25 inch) lites of conventional glass separated by air layers and up to two FEP films used as intermediate glazings. In each case the outdoor (cold side) glazing had a copper or gold coloured solar-control metal coating on the surface facing the other glazings. The individual panes of each glazing system were held at 11.94 mm (0.470 inch) spacings by spacers made of rigid insulating foam. A cross section of the edge of a double glazed prototype is shown in Figure 4.6. Thermal resistance values reported here were measured at the heater plate located at the centre of the warm copper plate where the heat transfer through the glazing system was assumed to be free of edge effects.

The hemispheric emissivities of the copper and gold coatings (0.04 and 0.132, respectively) were measured using a Gier-Dunkel DB-100 Infrared Reflectometer. This device provides a total, normal/hemispheric reflectance measurement with a claimed accuracy of 3% of full scale (full scale=1). The IR hemispheric optical properties of the FEP film (transmittance=0.624, absorptance=0.243) were based on the results reported by Wilkinson (21) who developed a procedure by which the capability of the DB-100 could be extended to make measurements on non-opaque films. This technique is described in reference (11). The conversion of DB-100 results to the desired hemispheric properties involves a small adjustment that is described in reference (22).

Each prototype system was simulated using a modified version of VISION. This program modification involved substituting the thermal resistance of the neoprene sheet, R_n, for the thermal resistance that would normally be accounted for between the glazing system and its environment. This alteration allows the program to simulate the indoor and outdoor temperature nodes, T_1 and T_n, as the hot and cold copper plates, respectively. The results of the simulation included the total heat flux through the glazing system, q_{tot}, plus the temperatures of the glazings adjacent to the hot and cold copper plates, T_2 and T_{n-1}, respectively. The glass-to-glass conductance, C_{VISION}, was then calculated as:

---

* FEP film is manufactured by E.I du Pont de Nemour and Co. under the registered trade mark Teflon.

$$C_{VISION} = q_{tot}/(T_2 - T_{n-1}) \tag{4.7}$$

The measured and calculated C-value results for the prototype glazing systems are shown in Table 4.1. In addition, all of the glazing systems have been simulated using VISION (in its unmodified form) and the U-values for ASHRAE winter and summer design conditions, U_w and U_s, respectively, as well as the shading coefficient (SC) are reported.

The measured and calculated C-value results shown in Table 4.1 agree very well. The largest discrepancies noted were 8% and were recorded for both of the double glazed systems tested at the higher temperature difference. In general, better agreement was found for glazing systems with more panes. In the cases where the measured C-value changed as a function of the applied temperature difference a similar change in C-value was generally noted in the simulation results. The discrepancy between calculated and measured C-values was 2% or less for 5 of the 10 tests carried out.

In a second study (23) four prototype glazing systems that incorporated low-e coatings and FEP films were constructed and tested. These glazing systems consisted of two panes of glass separated by either 0, 1, 2 or 3 FEP films used as intermediate glazings. In each case, the warm-side glass pane had a low-e coating on the surface facing the other glazing(s). The fourth glazing system had a second low-e coating on the cold-side glass pane - again on the surface facing the other glazings. The hemispheric emissivities of the low-e coatings (measured using the DB-100 Infrared Reflectometer) were 0.067 and 0.070.

Table 4.1
Comparison of Simulated and Measured C-Values;
Metal Solar Control Coatings

	Metal Coating	T_{cold} [°C]	# Teflon Films	C_{meas} [W/m^{2}K]	C_{VISION} [W/m^{2}K]	U_w [W/m^{2}K]	U_s [W/m^{2}K]	SC
1	gold	-10.0	0	2.72	2.94	1.95	2.91	0.10
2	gold	-10.0	1	1.95	1.89	1.41	1.65	0.09
3	gold	-10.0	2	1.63	1.62	1.24	1.50	0.08
4	copper	-10.0	0	2.32	2.51	1.78	1.80	0.16
5	copper	-10.0	1	1.64	1.56	1.23	1.37	0.15
6	copper	-10.0	2	1.38	1.35	1.08	1.25	0.15
7	gold	0.0	2	1.67	1.66	1.24	1.50	0.08
8	copper	0.0	0	2.31	2.35	1.78	1.80	0.16
9	copper	0.0	1	1.65	1.56	1.23	1.37	0.15
10	copper	0.0	2	1.40	1.37	1.08	1.25	0.15

Pane spacing = 11.94 mm, T_{hot} = 21.0 °C, slope = vertical, fill gas = air
$R_n = 0.017$ m^{2}K/W
T_{hot} and T_{cold} are constant temperature bath settings.

The measured and simulated C-value results for the low-e/FEP prototype glazing systems are shown in Table 4.2. Again, VISION was used to generate U_w, U_s and SC values which also appear in Table 4.2. The measured and calculated C-value results agree well. The largest discrepancy (13%) was noted for the fourth glazing system which was the glazing system with the greatest thermal resistance and the lowest heat flux. The agreement between measured and calculated results for the other glazing systems was much better. Figure 4.7 shows a cross section and various simulation results for the third glazing system described in Table 4.2. The total heat flux and its components are shown below the glazing system. It can be seen that the low-e coating is able to suppress radiative heat transfer in more than one gap but it is less effective in the more remote gaps.

Table 4.2
Comparison of Simulated and Measured C-Values;
Low-E Coating

	# Low E Coatings	T_{cold} [°C]	# Teflon Films	C_{meas} [W/m^{2}K]	C_{VISION} [W/m^{2}K]	U_w [W/m^{2}K]	U_s [W/m^{2}K]	SC
1	1	0.0	0	2.61	2.47	1.83	1.87	0.61
		-10.0		2.46	2.62			
2	1	0.0	1	1.69	1.64	1.25	1.45	0.59
		-10.0		1.67	1.61			
3	1	0.0	2	1.41	1.42	1.09	1.32	0.58
		-10.0		1.41	1.38			
4	2	0.0	3	0.93	0.83	0.72	0.82	0.47
		-10.0		0.93	0.82			

Pane spacing = 11.94 mm, T_{hot} = 20.0 °C, slope = vertical, fill gas = air
R_n = 0.017 m^{2}K/W
T_{hot} and T_{cold} are constant temperature bath settings.

DISCUSSION

An accurate method of measuring centre-glass heat flux in glazing systems has been described. The simplicity of the apparatus permits measurements to be carried out quickly and at low cost making this test useful as a developmental procedure. The measurement process provides a glass-to-glass thermal conductance and eliminates the problems involved in maintaining reproducible convective film coefficients as required by other test methods for glazing system U-value measurement. The apparatus permits testing glazing systems at any tilt angle, a feature not common to U-value test rigs.

Measured and calculated heat transfer results have been presented and compared for eighteen glazing systems incorporating FEP intermediate glazings used in conjunction with either low-e or metal solar-control coatings. The difference between calculated and measured results was as great as 13% for one glazing system but was less than 8% in all other cases and was 5% or less in more than half of the comparisons.

The concept of attenuating radiative heat transfer in more than one interpane gap by using a low emissivity surface adjacent to a highly IR transparent film was previously established as it applied to the design of solar collectors. This concept has now been experimentally demonstrated in the context of glazing systems built with FEP films and low-e or metal coatings.

REFERENCES

(1) Hollands, K.G.T., "Natural Convection in a Horizontal Thin-Walled Honeycomb Panels", Journal of Heat Transfer, Vol. 95, pp. 439-444, Nov. 1973.

(2) Hollands, K.G.T., Konicek, L., "Experimental Study of the Stability of Differentially Heated Inclined Air Layers", International Journal of Heat and Mass Transfer, Vol. 16, pp. 1467-1476, 1973.

(3) Hollands, K.G.T., Raithby, G.D., Konicek, L., "Correlation Equations for Free Convection Heat Transfer in Horizontal Layers of Air and Water", International Journal of Heat and Mass Transfer, Vol. 18, pp.879-884, 1975.

(4) Hollands, K.G.T., Unny, T.E., Raithby, G.D., Konicek, L., "Free Convective Heat Transfer Across Inclined Air Layers", Journal of Heat Transfer, Vol. 98, No. 2, 1976.

(5) ElSherbiny, S.M., Hollands, K.G.T., Raithby, G.D., "Nusselt Number Distribution in Vertical and Inclined Air Layers", Transactions of the ASME, Vol. 105, 1983.

(6) ElSherbiny, S.M., Hollands, K.G.T., Raithby, G.D., "Effect of Thermal Boundary Conditions on Natural Convection in Vertical and Inclined Air Layers", Journal of Heat Transfer, Vol. 104, 1982.

(7) Ruth, D.W., Hollands, K.G.T., Raithby, G.D., "On Free Convection Experiments in Inclined Air Layers Heated from Below", Journal of Fluid Mechanics, Vol. 96, pp.459-479, 1980.

(8) ElSherbiny, S.M., Raithby, G.D., Hollands, K.G.T., "Heat Transfer by Natural Convection Across Vertical and Inclined Air Layers", Journal of Heat Transfer, Vol. 104, Feb. 1982.

(9) ElSherbiny, S.M., Hollands, K.G.T., Raithby, G.D., "Free Convection Across Inclined Air Layers with One Surface V-Corrugated ", Transactions of the ASME, Vol. 100, 1978.

(10) Wright, J.L., Hollands, K.G.T., "Radiant and Free Convective Heat Transfer Through a Pleated (V-Corrugated) Teflon Film", Proceedings of the International Solar Energy Society Conference, Hamburg, 1987.

(11) Hollands, K.G.T., Raithby, G.D., Russel, F.B. Wilkinson, R.G., "Coupled Radiative and Conductive Heat Transfer Across Honeycomb Panels and through Single Cells", International Journal of Heat and Mass Transfer, Vol. 27, No. 11, pp. 2119-2131, 1984.

(12) Cane, R.L.D., Hollands, K.G.T., Raithby, G.D., Unny, T.E., "Free Convection Heat Transfer Across Inclined Honeycomb Panels", Journal of Heat Transfer, Vol. 99, no. 1, pp.86-91, 1977.

(13) Ford, C.J., "Convection Suppression in Flat Plate Solar Collectors by the Use of Square Celled Honeycombs", M.A.Sc. Thesis, University of Waterloo, Waterloo, Canada, 1987.

(14) Smart, D.R., Hollands, K.G.T., Raithby, G.D., "Free Convection Heat Transfer Across Rectangular-Celled Diathermanous Honecombs", Journal of Heat Transfer, Vol. 102, 1980.

(15) Hollands, K.G.T., Iynkaran, K., "Proposal for a Compound-Honeycomb Collector", Solar Energy, Vol. 34, No. 4/5, pp. 309-316, 1985.

(16) ElSherbiny, S.M.,. "Heat Transfer by Natural Convection Across Vertical and Inclined Air Layers", Ph.d. thesis, Department of Mechanical Engineering, University of Waterloo, Waterloo, Canada, 1980.

(17) Hollands, K.G.T., Wright, J.L., "Heat Loss Coefficients and Effective $\tau\alpha$ Products for Flat-Plate Collectors with Diathermanous Covers", Solar Energy 30(3), pp.211-216, 1982.

(18) Hollands, K.G.T., Wright, J.L., "Theory and Experiment on Heat Loss Coefficients for Plastic Covers", Proceedings of the American Section of the International Solar Energy Society, pp. 441-445, 1980.

(19) Wright, J.L., "The Computer Simulation of Super Window Glazing Systems which Incorporate Teflon (FEP) Inner Glazings," Proceedings of the ISES/ SESCI Intersol'85 Conference, Montreal, Canada, 1985, Vol. 1, pp. 267-271.

(20) Wright, J.L., Sullivan, H.F., "Simulation and Measurement of Windows with Metal Films Used in Conjunction with Teflon Inner Glazings", International Conference on Building Energy Management, Lausanne, Switzerland, September, 1987.

(21) Wilkinson, R.G., "Radiative-Conductive Heat Transfer Through a Honeycomb Panel", M.A.Sc thesis, Department of Mechanical Engineering, University of Waterloo, Waterloo, Canada, 1979.

(22) Wright, J.L., "Free Convection in Inclined Air Layers Constrained by a V-Corrugated Teflon Film", M.A.Sc. thesis, Department of Mechanical Engineering, University of Waterloo, Waterloo, Canada, 1980.

(23) Wright, J.L., Sullivan, H.F., "Simulation and Measurement of Windows with Low Emissivity Coatings Used in Conjunction with Teflon Inner Glazings," ISES Solar World Congress, Hamburg, West Germany, September (1987), Vol. 4, pp. 3136-3140.

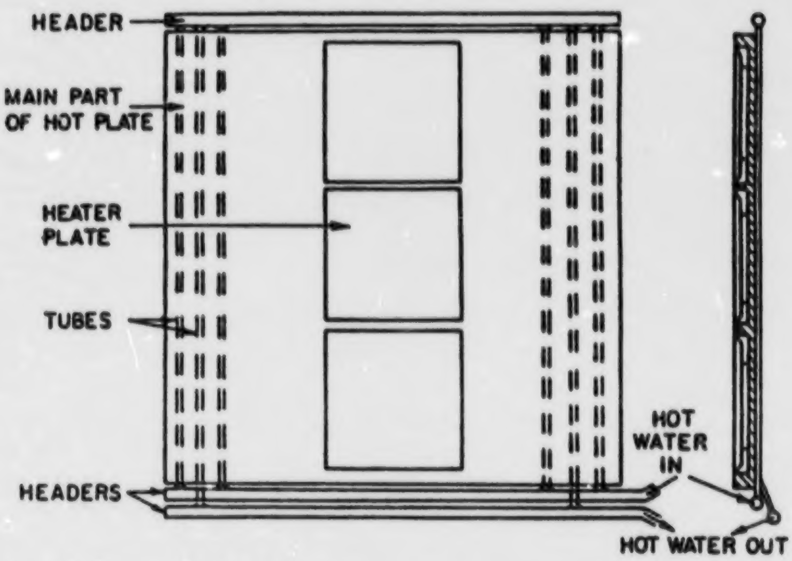

HEADER
MAIN PART OF HOT PLATE
HEATER PLATE
TUBES
HEADERS
HOT WATER IN
HOT WATER OUT

Figure 4.1: Sketch of the Hot Plate

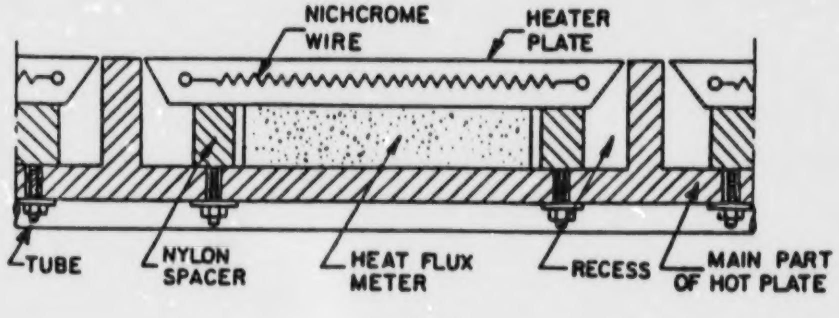

NICHCROME WIRE
HEATER PLATE
TUBE
NYLON SPACER
HEAT FLUX METER
RECESS
MAIN PART OF HOT PLATE

Figure 4.2: Heater Plate/Heat Flux Meter Detail

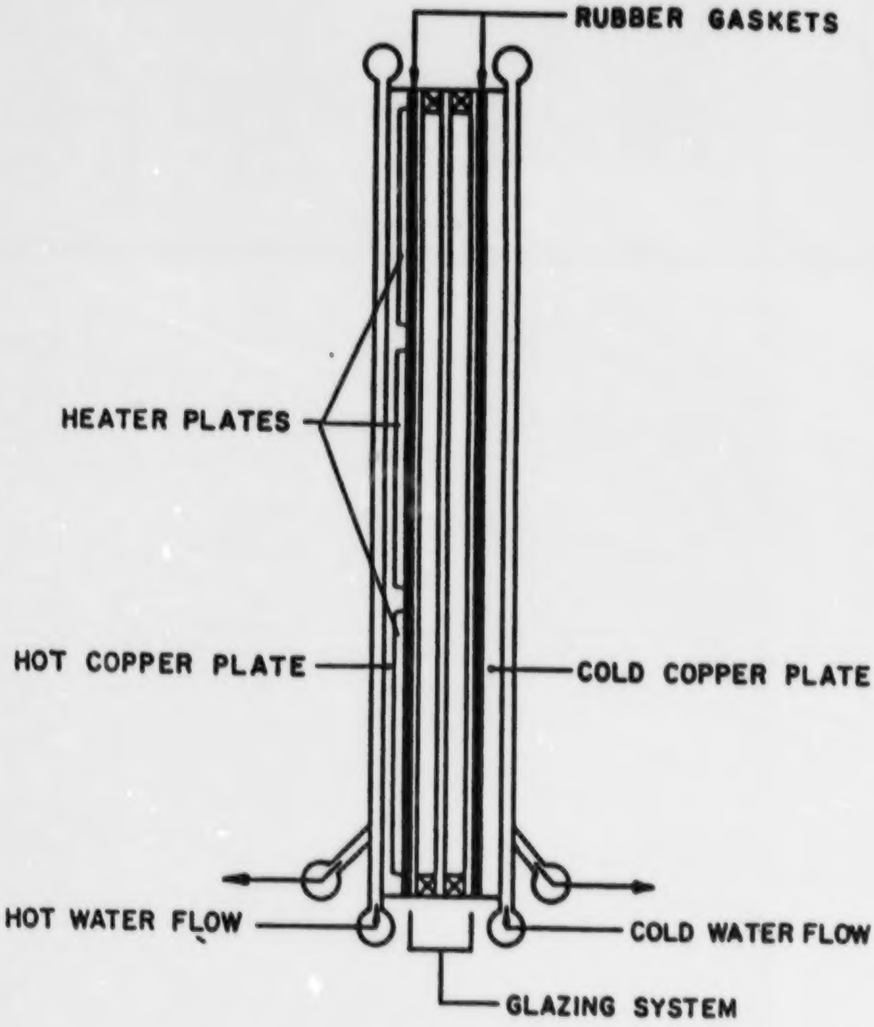

RUBBER GASKETS
HEATER PLATES
HOT COPPER PLATE
COLD COPPER PLATE
HOT WATER FLOW
COLD WATER FLOW
GLAZING SYSTEM

Figure 4.3: Copper Plates with Triple Glazed Test Unit

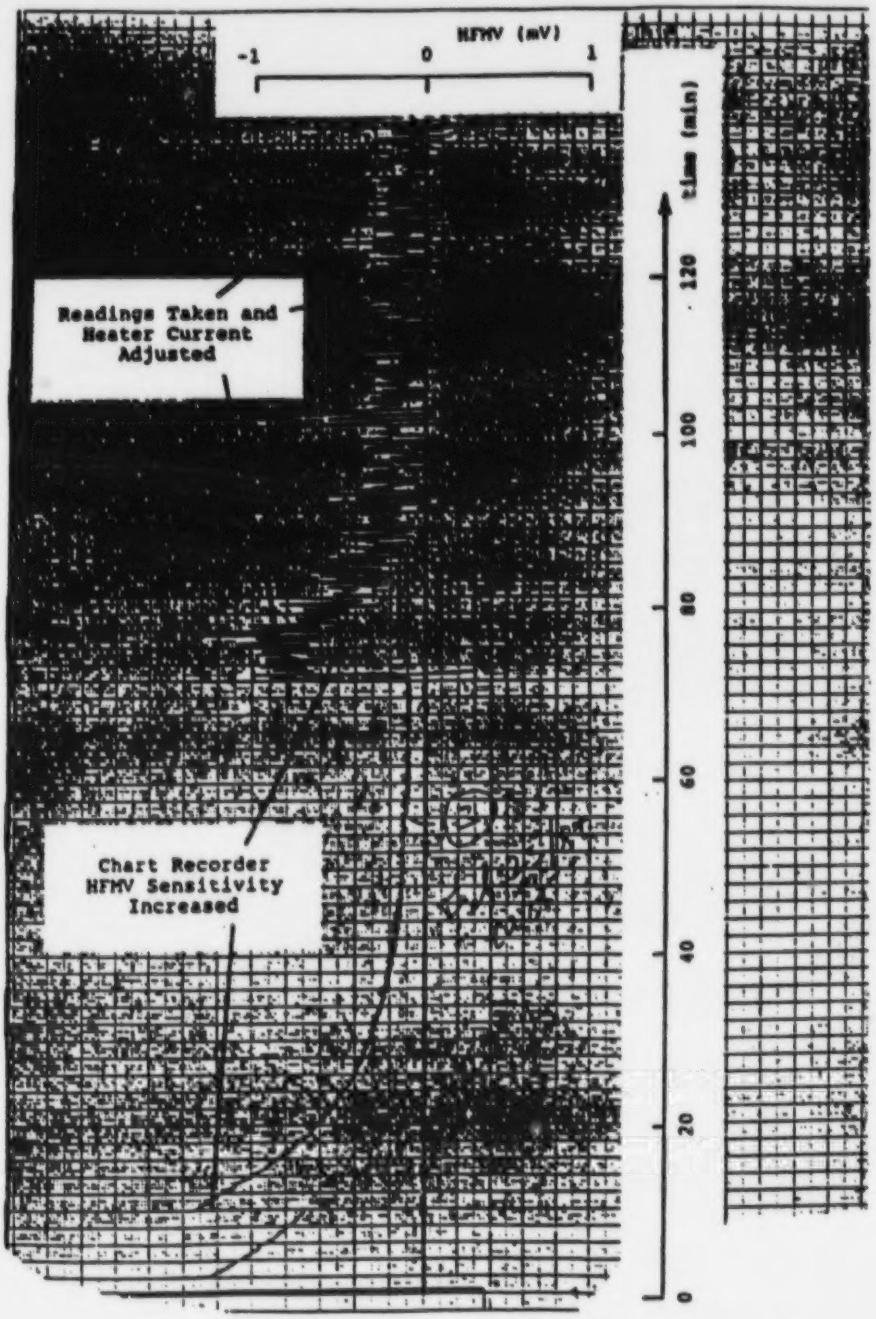


Figure 4.4: Sample HFMV Strip Chart Recorder Output

$HFMV_1$ (mV)	$HFMV_2$ (mV)	$HFMV_3$ (mV)	$V_{h,1}$ (V)	$V_{h,2}$ (V)	$V_{h,3}$ (V)	$I_{h,2}$ (A)	EMF (mV)	$Q_{e,2}$ (W)
0.46	0.16	-0.78	3.946	4.047	4.147	0.3873	7.180	1.567
0.20	0.10	-0.15	3.901	4.029	4.231	0.3856	7.182	1.554
0.00	0.00	-0.04	3.850	4.001	4.239	0.3829	7.180	1.532

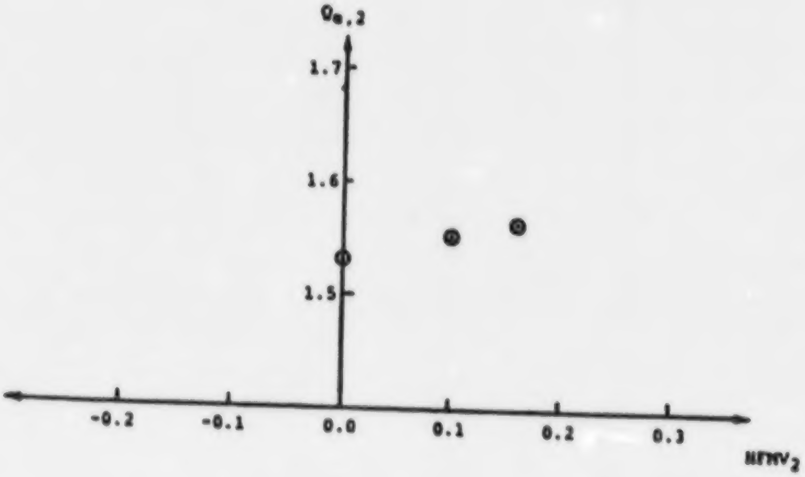


Figure 4.5 Sample Test Results and Q_e vs. HFMV Plot

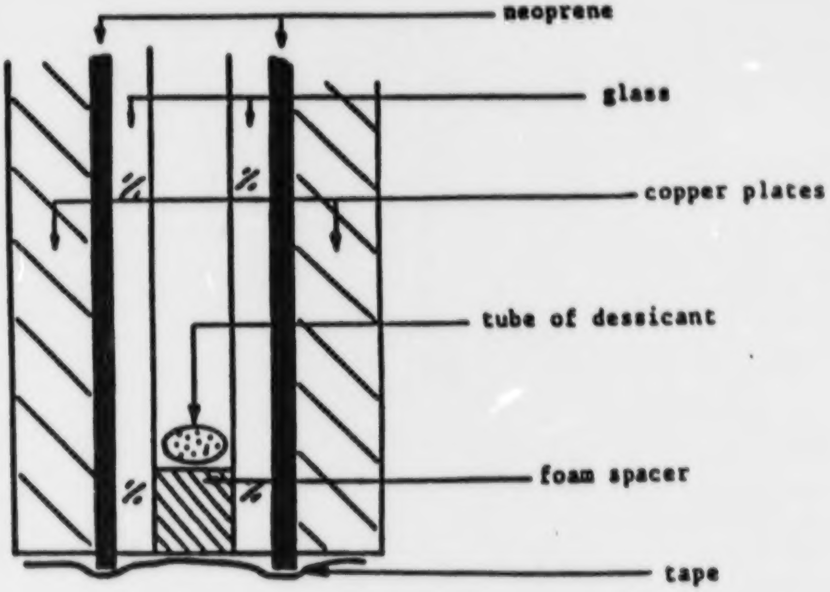

neoprene
glass
copper plates
tube of dessicant
foam spacer
tape

Figure 4.6: Cross Section at Edge of Double Glazed Test Unit

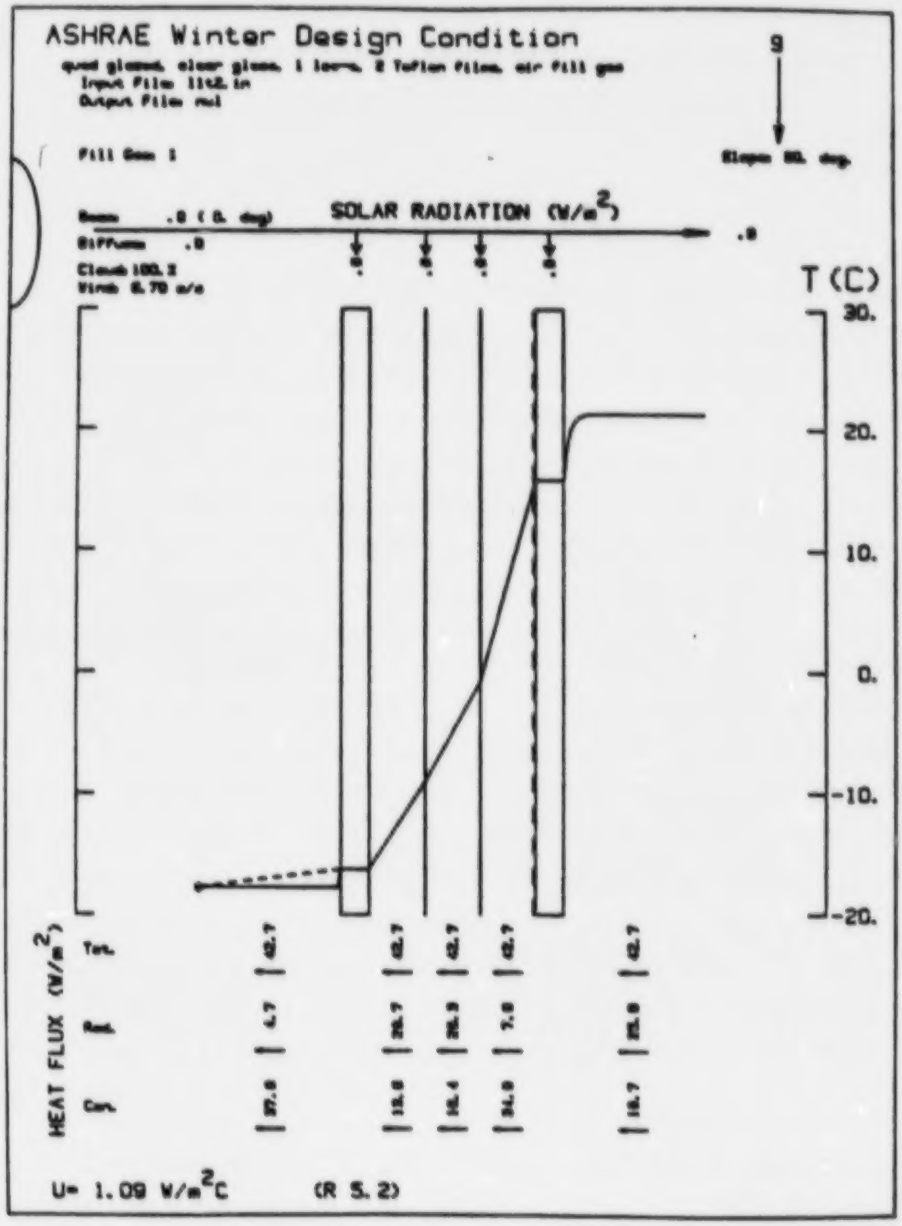


Figure 4.7: VISION Graphic Output Showing Centre-Glass Simulation Results for Glazing System GS3

Chapter 5

THERMAL RESISTANCE MEASUREMENT OF GLAZING SYSTEM EDGE-SEALS

INTRODUCTION

A typical glazing assembly consists of two sheets of glass separated at their edges by an edge-seal assembly. Advances in glazing system technology, primarily low-emissivity coatings and low conductivity fill gases, have substantially increased the thermal resistance of commercially available glazing systems. The resulting increase in centre-glass thermal resistance has highlighted the problems caused by the thermal short-circuit associated with edge-seal conduction. New and more innovative edge-seal designs are being developed as a result of this problem.

A sketch of the most widely used edge-seal design is shown in Figure 5.1. The construction consists of two glass sheets, a spacer bar plus primary and secondary sealants. Sealants are usually chosen so that the primary sealant has a very low water vapour transmission (polyisobutylene (PIB) is common) and the secondary sealant provides structural integrity (frequently polysulphide or butyl[+]). Some designs do not incorporate a primary sealant between the spacer bar and glass surfaces and are known as single-seal edge-seals. Edge-seals that incorporate both the primary and secondary sealants are called double-seal edge-seals.

In the present study, a guarded heater plate apparatus, the University of Waterloo Natural Convection Apparatus (UWNCA), was used for the direct measurement of heat transfer through a series of edge-seal test samples (1,2). Thermal resistance values were measured for nine test samples. These results are useful in that they provide reliable

---

[+] Butyl is a generic name for a class of tar-like sealants used in edge-seals. Many different manufacturers have formulated their own versions. If a particular butyl must be heated before application it is called hot-melt butyl.

estimates of edge-seal thermal resistance values used as input to subsequent 2-D glazing system simulations. These data also provide a new perspective regarding the relative thermal performance levels to be expected from various edge-seal designs as well as an insight into the way in which edge-seals conduct thermal energy. The literature contains no record of other edge-seal thermal resistance measurements.

TEST PROCEDURE

Table 5.1 presents a list of the spacer bar/sealant combinations for which edge-seal test samples were built and tested. Figure 5.2 shows how the more conventional edge-seal test samples containing aluminum or fibreglass spacer bars were constructed. A two part silicone was used in most cases because it could readily be gunned into the long narrow passages of the edge-seal test samples. The designs listed in Table 5.1 correspond closely to the spacer bar/sealant configuration shown in Figure 5.1 with the exception of samples 3, 5 and 6. Figure 5.3 shows how the currugated strip spacer (sample number 3) would be installed in an SGU. This edge-seal consists of single corrugated aluminum strip (with the appearance of a continuous sine wave) embedded in butyl. The corrugated strip edge-seal test sample was built by placing a series of seal strips side-by-side in a continuous layer between two sheets of glass. The layout of samples incorporating foam spacers (samples 5 and 6) is shown in Figure 5.4. All of the edge-seals and edge-seal components are commercially available.

Thermal resistance testing was carried out using the UWNCA guarded heater plate apparatus. The test samples (each 305x305 mm (12 x 12 inches)) were placed between these plates but separated from the plates by neoprene mats. The steady-state heat transfer through each sample (driven by the temperature difference between the plates) was measured over the face of the centre guarded heater plate. The measured heat transfer rate, plate-to-plate temperature difference and known thermal resistance of the neoprene mats was combined to give a measured thermal resistance of the test sample.

Table 5.1
List of Edge-Seal Test Samples

Sample no.	Spacer Bar	Secondary Sealant	Primary Sealant	Pane Spacing mm (in)
1	Aluminum	Silicone (g)+		13.1 (0.516)
2	Aluminum	Silicone (g)	PIB*	14.0 (0.551)
3	Corrugated Strip	Butyl	Butyl	14.1 (0.555)
4	Fibreglass	Silicone (b)+	Silicone (b)	12.7 (0.500)
5	Foam!	Hot-Melt Butyl		13.1 (0.516)
6	Foam!	Silicone (g)		12.9 (0.508)
7	Fibreglass	Silicone (g)		12.6 (0.496)
8	Fibreglass	Silicone (g)	PIB	12.8 (0.504)
9	Aluminum	Silicone (g)	PIB	13.0 (0.512)

+ The silicone sealant used is designated as either (g) or (b) depending upon whether its colour was grey or black.

* PIB is an abbreviation of Polyisobutylene.

! This edge-seal consists of approximately 50% foam spacer and 50% sealant.

Each edge-seal test section was constructed and tested as shown in Figure 5.2. The spacer bars were arranged and back-filled with sealant in order to closely reproduce the construction of commercially produced sealed glazing units. Alternate seals were placed back-to-back such that planes of symmetry between each pair of seals could be considered adiabatic. This was done to ensure that the heat flux, q, measured over the area of a heater plate, would be an accurate measure of the heat flux through each individual seal.

The measured quantities of prime importance from the solid seal experiments were the plate-to-plate temperature difference, ΔT_{pp}, and the heat flux between the plates, q. The ratio of these two values provides the total thermal resistance of the neoprene, glass and seal layers. The thermal resistance of the seal layer alone, R_{seal}, was estimated by subtracting the resistance of the neoprene mats ($2R_n$) and the sheets of glass ($2t_g/k_g$). This representation of thermal resistance, expressed by Equation 5.1, is a direct measure of resistance to heat transfer provided by the edge-seal on a "per unit area" basis.

$$R_{seal} = (\Delta T_{pp}/q) - 2R_n - 2(t_g/k_g) \qquad (5.1)$$

k_g = glass conductivity (0.96 W/m·C (3))
t_g = glass thickness

In order to provide a more useful representation of the results, the measured thermal resistances were recast into the form of a thermal conductance on a "per unit length of seal" basis, k_{lin}. This linear conductance is defined by Equation 5.2.

$$Q = q \cdot A = A \cdot (T_1 - T_2)/R_{seal} = L_{seal} \cdot k_{lin}(T_1 - T_2) \qquad (5.2)$$

Q = heat loss through seal of length L_{seal}
A = area of seal in contact with glass = $L_{seal} \cdot w_{seal}$
w_{seal} = width of an individual seal (see Figures 5.2 and 5.4)
$T_1 - T_2$ = temperature drop through seal layer

Equation 5.2 can be rearanged to give:

$$k_{lin} = w_{seal}/R_{seal} \qquad (5.3)$$

RESULTS

Table 5.2 presents a summary of the measured results for the nine edge-seal test sections. In each case the measured values of ΔT_{pp}, q, t_g and w_{seal} are shown along with the resulting values of R_{seal} and k_{lin}. The linear conductance results of Table 5.2 are presented graphically in Figure 5.5.

Table 5.2
Summary of Results for Edge-Seal Test Units

		ΔT_{pp} [°C]	q [W/m^2]	t_g [mm]	w_{seal} [mm]	R_n [m^{2}K/W]	R_{seal} [m^{2}K/W]	k_{lin} [W/m·K]
1	Aluminum SS*	18.70	372.2	3.96	12.8	0.017	0.008	1.6
2	Aluminum DS*	18.78	383.9	3.96	12.9	0.017	0.007	1.9
3	Corrug'd Strip	18.90	324.8	2.90	7.43	0.017	0.018	0.41
4	F'glass	19.18	192.2	2.84	16.3	0.017	0.060	0.27
5	Foam/ Hot-Melt Butyl	18.98	187.9	4.75	12.3	0.009	0.073	0.17
6	Foam/ Silicone	18.69	302.15	3.91	12.7	0.009	0.036	0.36
7	F'glass SS	18.95	272.5	3.91	12.7	0.009	0.043	0.29
8	F'Glass DS	18.96	277.9	3.91	12.7	0.009	0.042	0.30
9	Aluminum DS	18.41	574.3	3.91	12.6	0.009	0.006	2.1

* SS and DS are abbreviations of single-seal and double-seal.

One note of caution is in order. While testing units with very low thermal resistance (e.g., units 1 and 2) the majority of the thermal resistance measured between the copper plates was due to the neoprene mats. The accuracy of the measured thermal resistance of the seal is less than in the experiments where thinner mats were incorporated or where the seals provided more thermal resistance. It is safe to say that the single-seal (unit 1) provided more thermal resistance than the double-seal (unit 2), but to say it had 19% more thermal resistance would be unfounded. The important observation is that the thermal resistance between the two sets of seals (with and without the conventional aluminum spacer bar) differed by a significant amount. This distinction could make the difference between having or not having condensation on windows in the winter time. The results of other experiments where thinner neoprene mats were used (R_n=0.009 m^2K/W; see Table 5.2) are much less likely to be in error.

DISCUSSION

The data shown in Figure 5.5 demonstrate that the seals tested possess a wide range of k_{lin}; a factor of 12 exists between the lowest and the highest. These seals can readily be split into two groups on the basis of their thermal performance. The more conventional single and double-seals (units 1,2 and 9) provide little thermal resistance while the edge-seals incorporating corrugated metal, fibreglass or foam spacers provide considerably more thermal resistance. The greatest thermal resistance was measured across the foam edge-seal with hot-melt butyl sealant.

The results from these edge-seal tests reveal that the single-seal configurations provided more thermal resistance than similar double-seal configurations in the two cases where direct comparisons could be made. Compare the k_{lin} results for unit 1 versus units 2 and 9 which all had aluminum spacer bars and silicone edge sealant or examine the results for unit 7 versus unit 8 which both had fibreglass spacer bars and silicone edge sealant. The thermal resistance of edge-seals with aluminum spacer bars seems to be sensitive to the placement of sealant between the spacer and the glass. It might be reasoned that a large portion of the thermal resistance was due to the material in place between the aluminum bar and the glass. Further reasoning indicates that some of the thermal resistance present in the single-seal configuration resulted from a contact resistance between the spacer and the glass and that this resistance was reduced by the presence of the primary sealant in the double-seal design. The results for single and double-seal edge-seals with fibreglass spacers (units 7 and 8) support this line of reasoning. In this case, much more thermal resistance exists in the spacer bar and k_{lin} is insensitive to the presence of primary sealant between the spacer and the glass.

The thermal performance of the edge-seal that incorporates the foam spacer is highly sensitive to the choice of edge sealant that is used. Compare k_{lin} for units 5 and 6 where the use of silicone sealant instead of hot-melt butyl approximately doubled the linear conductance of the edge-seal. Clearly, the majority of the heat transfer occurs through the sealant rather than the spacer as opposed to edge-seals with aluminum spacers. The thermal conductivity figures shown in reference (2) support this assertion in that the conductivity of silicone was found to be three times higher than the conductivity of the foam. It can be reasoned that the thermal resistance of edge-seals incorporating the fibreglass spacer would also be sensitive to the conductivity of the edge sealant but to a lesser extent because the fibreglass edge-seal design includes a slightly smaller portion of sealant (about 30% sealant versus 50% for the foam design). It is likely that if a fibreglass edge-seal with hot-melt butyl sealant had been tested it would have had significantly more thermal resistance than the samples that were tested with silicone edge sealant (units 4, 7 and 8). On the other hand, the use of hot-melt butyl instead of silicone would likely have had little impact on the performance of the units with aluminum spacer bars.

It is instructive to consider the homogeneous edge-seal configuration shown in Figure 5.6. In this case the cross-section of the edge-seal has the dimensions w_{seal} and t_{seal}, the latter being equal to the pane spacing. An edge-seal test sample for this construction would simply contain a uniform slab of the seal material. The heat flux expected through the test section is given by Equation 5.4.

$$q = (k_{seal}/t_{seal})\cdot(T_1 - T_2) \qquad (5.4)$$

k_{seal} = thermal conductivity of seal material
t_{seal} = seal thickness (equal to pane spacing)
$T_1 - T_2$ = temperature drop through seal layer[*]

This result combined with Equation 5.2 yields:

$$k_{lin} = k_{seal}\cdot(w_{seal}/t_{seal}) \qquad (5.5)$$

Thus, for an edge-seal made of a homogeneous material the linear conductance can be estimated knowing only the thermal conductivity of the seal material and the aspect ratio of the seal, w_{seal}/t_{seal}.

Examine the homogeneous edge-seals shown in Figure 5.7a. One seal is twice the size of the other but they would have the same linear conductance. The seals shown in Figure 5.7b differ only in that the one on the right provides twice as much pane spacing. Consequently, it has half the linear conductance of the seal on the left.

Consider the corrugated strip edge-seal (test sample no. 3). The aspect ratio of this seal is approximately equal to 1/2 (w_{seal}/t_{seal} = 7.43/14.1). Assume it is known that the thermal conductivity of the butyl used in this seal is k_{seal}=0.2 W/m·K. If this seal were built without the internal aluminum strip then a linear conductance value of k_{lin}=0.1 W/m·K would be expected according to Equation 5.5. In this case the measured value of k_{lin}=0.41 W/m·K suggests that there is potential for significant improvement if a metal with lower conductivity or thinner cross-section (or both) were substituted.

If an edge-seal were constructed simply by bonding a metal strip or foil to the outer edges of two sheets of glass its linear conductance could be estimated using Equation 5.5. Furthermore, if a particular metal were chosen for this application it would be possible, by rearanging Equation 5.5, to determine the maximum thickness of the metal strip, w_{max}, that would provide a specified value of k_{lin}:

---

[*] Each seal/glass interface is assumed to be isothermal for the purpose of simplifying the analysis. This is expected to be a good approximation (especially in the case of seals with high thermal resistance) because of the very high conductivity of the copper plates.

$$w_{max} = t_{seal} \cdot (k_{lin}/k_{seal}) \tag{5.6}$$

For example, say the desired edge-seal conductance is k_{lin}=0.5 W/m·K and the pane spacing is to be 12.7 mm. The calculated thicknesses for copper, aluminum and stainless steel edge-seals (k=400, 237 and 17 W/m·K, respectively) are w_{max}=0.016, 0.027 and 0.374 mm, respectively.

CONCLUSIONS

The seals tested can be grouped into two sets. Seals providing low thermal resistance were the single and double-seals with conventional aluminum spacer bars. Seals with high thermal resistance incorporated corrugated metal, fiberglass or foam spacers. The difference in thermal resistance between these two groups of seals was significant. The presence of a primary sealant in the double-seal design appears to lower the edge-seal thermal resistance - more for the edge-seals with aluminum spacers and very slightly for the fibreglass spacers. The conductivity of sealant used with the higher thermal resistance spacer bars (fibreglass and foam) has a strong bearing on the thermal resistance of the complete edge-seal.

The guarded heater plate measurements of the edge-seal test samples have provided a direct measure of the edge-seal thermal resistance. In addition, these data can readily be used as input to the detailed 2-D computer simulation of glazing systems.

REFERENCES

(1) Wright, J.L., "Thermal Resistance Measurement of Glazing System Edge-Seals," Second National Conference on Energy Efficient Cooling, Bethesda, Maryland, November 9,10 (1989).

(2) Wright, J.L.; Sullivan, H.F., "Thermal Resistance Measurement of Glazing System Edge-Seals and Seal Materials Using a Guarded Heater Plate Apparatus," ASHRAE summer meeting, Vancouver, (1989), preprint VA-89-11-3.

(3) Bolz, R.E., Tuve, G.L., "CRC Handbook of Tables for Applied Engineering Science", 2nd Edition, Page 177, CRC Press, 1970.

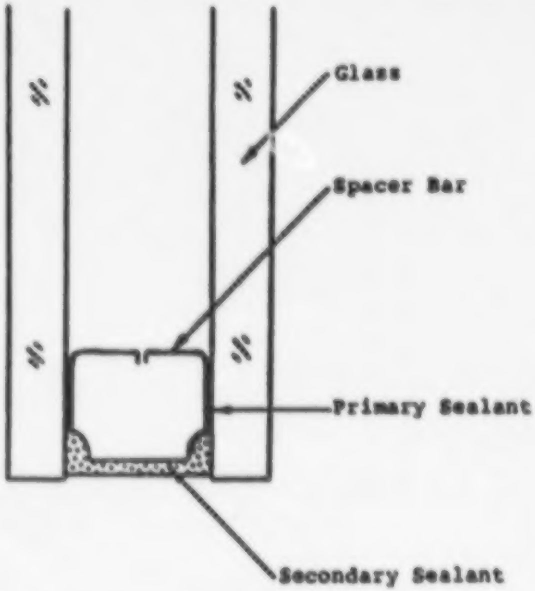
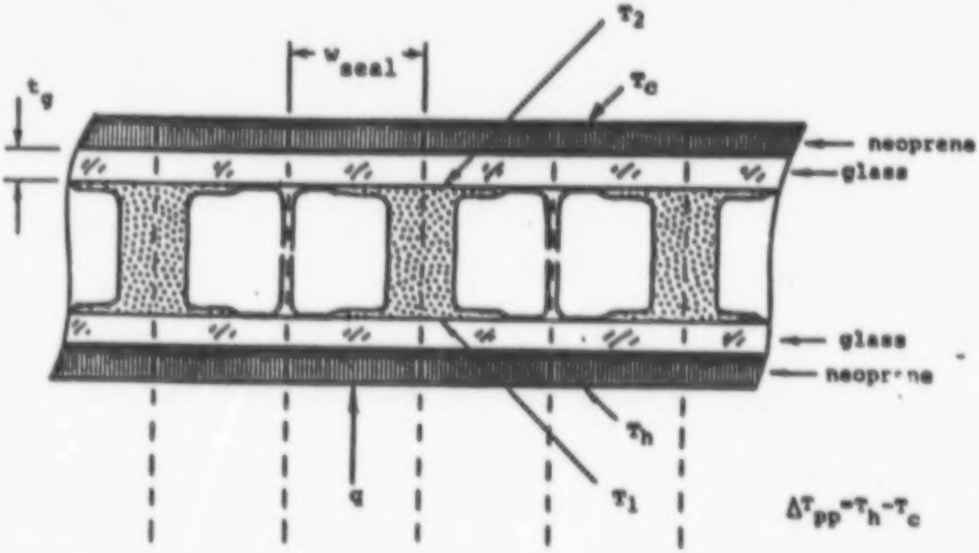

Glass
Spacer Bar
Primary Sealant
Secondary Sealant

Figure 5.1: Typical Edge-Seal Design


T_2
w_{seal}
t_g
T_c
neoprene
glass
glass
neoprene
T_h
q
T_1
$\Delta T_{pp} = T_h - T_c$

Dashed lines mark planes of symmetry.

Figure 5.2: Cross-Section of Edge-Seal Test Sample

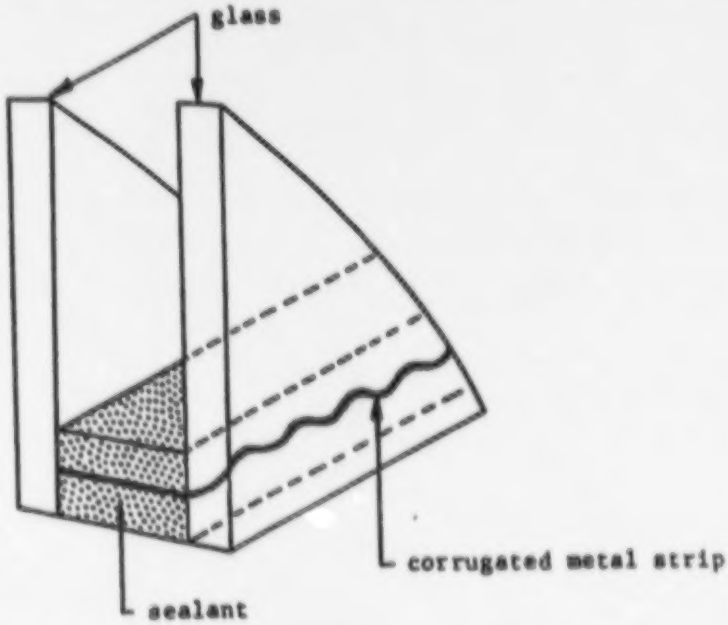
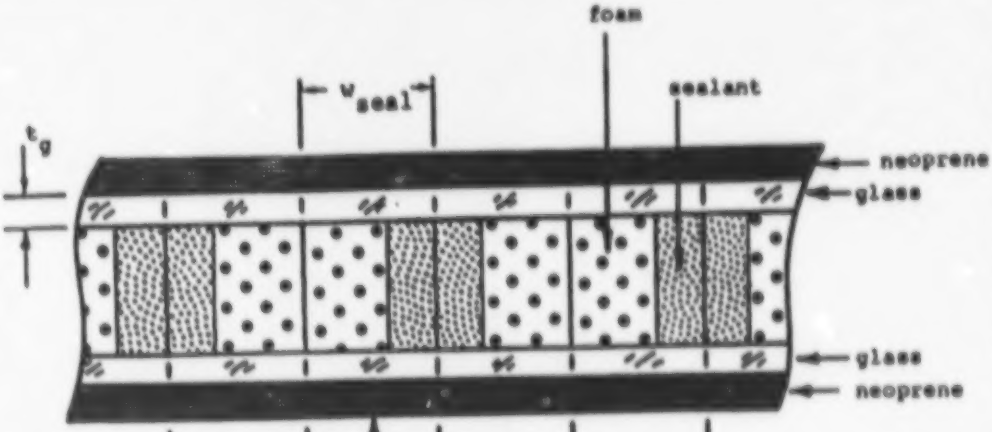

glass
corrugated metal strip
sealant

Figure 5.3: The Corrugated Strip Edge-Seal Design


foam
w_{seal}
sealant
t_g
neoprene
glass
glass
neoprene
q

Dashed lines mark planes of symmetry.

Figure 5.4: Cross-Section of Edge-Seal Test Sample with Foam Spacer

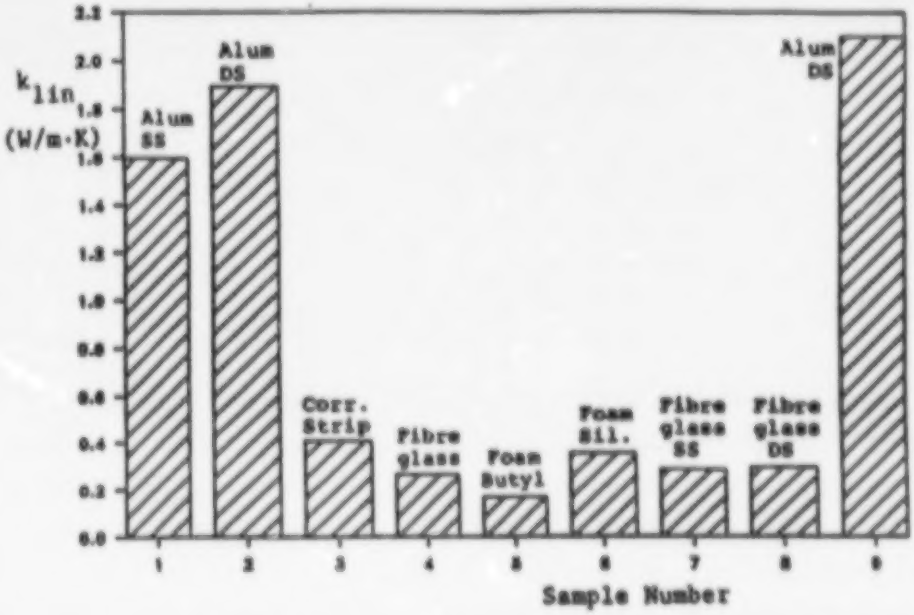


Figure 5.5: Measured Linear Conductance Results

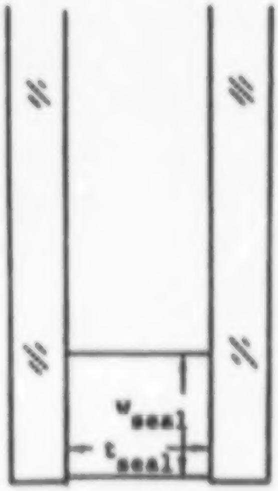

w
seal
t
seal

Figure 5.6: Homogeneous Edge-Seal Dimensions

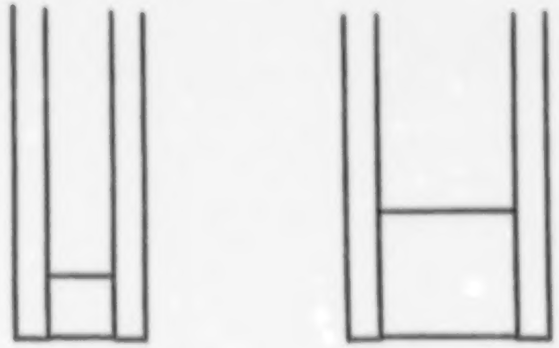


Figure 5.7a: Examples of Homogeneous Edge-Seal Configurations, Equal Linear Conductance Values

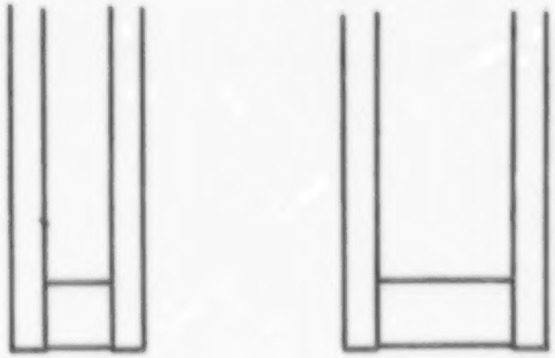


Figure 5.7b: Examples of Homogeneous Edge-Seal Configurations, Unequal Linear Conductance Values

Chapter 6

NATURAL CONVECTION IN A VERTICAL RECTANGULAR SLOT

REVIEW

Background

Most windows manufactured today contain a glazing system that is packaged in the form of a sealed glazing unit (SGU). The SGU consists of two panes of glass that are separated from each other by an edge-seal. The edge-seal isolates the cavity between the glazings - thereby reducing the number of surfaces to be cleaned and creating an insulating cavity suitable for non-durable low-e coatings and/or substitute fill gases. In contrast to the glazing system, few options are commercially available to increase the thermal resistance of the SGU edge-seal. Design improvements have dealt mainly with the requirements of the edge-seal to exclude moisture, provide a desiccant for the sealed space and retain the structural integrity of the SGU. Hence, the thermal bridge created by the edge-seal results in a band at the perimeter of the SGU where the temperature of a glazing can vary significantly as a function of distance from the edge of the glazing. This is an area of increased thermal stress in the glass (1), high energy loss and the site of condensation in the winter.

During cold weather, the convective flow of fill gas within the sealed space of an SGU is such that it contributes to the condensation problem at the bottom edge of the indoor glazing. Fill gas in the sealed space flows upward near the indoor glazing and downward near the outdoor glazing. The descending gas becomes progressively colder. At the bottom of the cavity this cold fill gas turns and comes in direct contact with the bottom of the indoor glazing where it starts its ascent. Thus, the glass near the bottom edge of the indoor glazing is cooled by the coldest fill gas in the interpane gap. A similar situation occurs at the top of the cavity where the fill gas heats the top of the outdoor glazing.

Experimental results support the idea that fill gas motion contributes to the bottom-edge condensation problem. Heat flux measurements using a guarded heater plate apparatus (2) have consistently shown that the heat flux to the bottom of the warm side glazing is higher than the heat flux to the top of the same glazing. Clearly, any model attempting to quantify local heat transfer rates in these regions or intended to determine the temperature distribution across the face of the glazing must account for both the edge-seal heat loss and the natural convection of the fill gas.

Edge-Glass Heat Transfer

Two calculation methods for estimating edge-glass heat transfer have recently been devised. One method has resulted from a joint effort by researchers in Switzerland, Belgium and France as part of the Windows and Fenestration Task of the International Energy Agency (IEA) Annex XII (Energy Conservation in Buildings and Community Systems Programme). IEA workers have proposed a set of edge-seal conductances or "linear" conductance values (3) based on measurements and finite-difference calculations. The recommended linear conductance values are shown in Figure 6.1. The centre-glass U-value is applied over the entire glazed area and additional heat loss caused by the edge-seal is calculated as the product of the linear conductance, length of edge-seal and the indoor/outdoor temperature difference. The reports resulting from this IEA task (4-9) deal with a wide range of topics concerning fenestration. Information regarding the estimation and use of linear conductance values is contained in reference (6). A similar set of linear conductance values being considered by the ISO working group on the Thermal Transmision Properties of Windows provides more detail in that edge-seals are categorized as being metal or non-metal (10).

The second method of edge-glass U-value calculation is now an ASHRAE procedure (11) and is based on unpublished hot-box results from a major manufacturer which were verified by a limited number of calculations. This procedure uses prescribed area ratios of frame, edge-glass and centre-glass with the edge-glass U-value being determined as a function of the centre-glass U-value. For example, the edge-glass U-value for a standard double glazed system with conventional metal edge-seal is 1.2 times greater than the centre-glass U-value. In the case of the same system with a soft low-e coating the suggested edge/centre U-value factor is 1.5.

The edge-glass calculations outlined above are useful for estimating the thermal losses of windows but they give no information about the variation in heat flux to be expected between the top and bottom edges of the window. They cannot provide temperature profiles of individual glazings.

Two 2-D finite-difference computer programs exist that are specifically designed for the analysis of heat loss through window frames. One program was developed in Sweden by Jonsson (12) and the other by Standaert (13) in Belgium. A third program based on the work of Jonsson has been produced by Carpenter (14,15). A sample of the graphic output taken from the work of Carpenter is shown in Figure 6.2. These window frame analysis programs calculate a temperature and heat flux solution through the edge-glass area of the glazing system. Neither the fill gas flow nor the 2-D nature of the radiative heat transfer are included in these simulation procedures. Heat transfer through the SGU is approximated by treating the sealed cavity as though it were filled with a solid material that is opaque to thermal radiation. This fictitious material is assigned an "effective" thermal conductivity that is determined as a function of the total centre-glass heat flux. The absence of fill gas flow is apparent in the symmetry of the isotherms in the fill gas near the end of the glazing cavity. The minimum temperature near the bottom of the indoor glazing will be consistently overpredicted as a result of the simplified fill gas model.

Natural Convection Between Glazings

The 2-D analysis of natural convection in the interpane cavity requires treatment of fill gas flow in a tall, vertical, rectangular slot. The fill gas is heated by one of the vertical walls and is cooled by the other. The wall temperatures are not uniform - with the most pronounced variations occurring near the edge-seals. Similarly, the conditions at the horizontal surfaces are not simple and cannot be specified as having zero heat flux (ZHF) or a linear temperature profile (LTP).

The literature contains an abundance of information about rectangular cavities where a temperature difference between the vertical walls drives a convective flow. The geometry and some of the nomenclature are shown in Figure 6.3. It has been shown that the solution is a function of the Rayleigh number, Ra, the aspect ratio of the cavity, A, and the Prandtl number of the fluid, Pr. Relatively few papers deal with conditions of interest in glazing units: $A \geq 40$, for gases $Pr \approx 0.71$ and for air and argon $Ra < 1.2 \times 10^4$. Furthermore, these studies almost universally prescribe isothermal side walls[+] and either LTP or ZHF boundary conditions. Nonetheless, it is useful to review the earlier results in that information is available concerning variables that affect the fill gas flow, the various flow regimes, instabilities in the flow, conditions under which certain flow regimes occur and details concerning effective modeling.

Nusselt first reported heat transfer measurements for this problem in 1909. Many authors have provided additional information (including (16-84)). Some studies (16-39) have suggested empirical relationships for the average heat flux over the vertical cavity wall (expressed as a Nusselt number, $Nu = Nu(Ra, Pr, A)$). Most of these correlations cannot be applied to the current problem in that they are not strictly valid for the desired range of Ra (17,21,24,28,32,38), Pr (25,27,29) or A (17,21,24-27,29-31,33,34,37-39). Some researchers either neglected or did not discern the dependence of Nu on A (19,24,26,32,38). Several correlations remain - the most suitable one being that of ElSherbiny, Raithby and Hollands (16) because it is based on a well established experimental procedure carried out over very wide ranges of Ra and A with the specific aim of independently resolving the roles of Ra and A. LTP conditions were closely approximated in these experiments. The Nu vs. Ra data of ElSherbiny, Raithby and Hollands

---

[+] Only one study was found (33) where side walls were not isothermal. Simulation of an SGU was performed with glazing temperature profiles based on hot-box measurements. Computed and measured local heat transfer rates did not agree well.

(vertical cavity) are shown in Figure 6.4. The solid lines plotted in Figure 6.4 represent the approximate method of Raithby, Hollands and Unny (35).

Batchelor (20) analysed the laminar natural convection and was the first to define conduction and boundary layer flow regimes. Eckert and Carlson (22) quantified local heat transfer using an interferometer and refined Batchelor's work by proposing conduction, transition and boundary layer regimes. These flow regimes can be understood by considering the flow at the mid-height of the cavity with Pr and A held constant. Figure 6.5 (based on data from reference (78)) shows computed profiles of the vertical velocity component and temperature for three values of Ra. When a small temperature difference is applied across the air layer (see $Ra=10^3$), a weak unicellular flow exists and the temperature profile across the cavity is linear. Heat transfer across the cavity takes place primarily by conduction (except in small regions at the ends of the cavity) with the result that $Nu=1$. This is called the conduction regime.

When the temperature difference is increased (see $Ra=10^4$ and 10^5) the flow strengthens and pulls closer to the walls in the form of two increasingly independent boundary layers. Elder (40) and Gill (42) pointed out that the boundary layer thickness is proportional to $Ra^{-1/4}$. At higher values of Ra the boundary layers become more distinct and are separated by a core region with the heat transfer taking place more by convection via the boundary layers and less by conduction across the core. Heat transfer across the cavity is greater than in the conduction regime ($Nu>1$).

In the conduction regime no vertical temperature gradient exists within the core but a vertical temperature gradient does exists within the core once the flow leaves the conduction regime. This temperature gradient is approximately linear with height, except near the ends of the cavity, and creates a stable stratification of the core fluid. The presence of stratification in the core can be used as a means of delimiting the conduction regime. More frequently, the nature of the fluid flow is categorized using the non-dimensional horizontal temperature gradient at the mid-point of the cavity, β_h $(\beta_h=-(\partial T/\partial x)/((T_h-T_c)/\ell_x)$. The conduction regime is characterized by $\beta_h=-1$, the boundary layer regime by $\beta_h \geq 0$ and the transition regime by $-1<\beta_h<0$. The three curves shown in the lower portion of Figure 6.5 are typical of temperature profiles for each of these three laminar flow regimes. A good discussion regarding the balance between shear and buoyant forces occuring in the conduction and boundary layer regimes is given by Raithby, Hollands and Unny (35).

The critical value of Ra at which flow leaves the conduction regime lies in the range $10^3<Ra<6x10^3$ and is a function of A. This aspect ratio dependence can be seen in Figure 6.4. The convective flow leaves the conduction regime at lower values of Ra in cavities with lower values of A.

If Ra is increased sufficiently, instabilities occur that create time dependent flow and eventually a turbulent boundary layer flow. The transition from laminar to turbulent flow can readily be pinpointed in the approximate method of Raithby, Hollands and Unny shown in Figure 6.4. Turbulent flow is represented by the line that extends upward to the right with a slope of 1/3. The lines inside the knee created by the turbulent boundary layer line and the horizontal axis have a slope of 1/4 and represent laminar boundary layer flow for various values of A. The critical value of Ra for the onset of turbulent flow is a function of A. The flow in enclosures with larger aspect ratios becomes turbulent at smaller values of Ra. The results of this theory suggest that the flow in very tall, narrow slots can become turbulent directly from the conduction regime.

The experimental data shown in Figure 6.4 display some trends that are similar to those of the theory. However, they do not show the ordered progression (as a function of A) that might be expected inside the knee. The measured Nu vs. Ra curves are tightly grouped and for $A>20$ they all depart the conduction regime at $Ra \approx 6x10^3$. It is tempting to conclude, on the basis of the similarity between the shapes of the measured and theoretical curves, that the flow immediately enters the turbulent regime. However, it is not clear whether Nu increases because the flow enters the laminar boundary layer regime, because it becomes turbulent or because of some other phenomenon.

A clue regarding the nature of the flow at $Ra>6x10^3$ can be taken from the work of Yin, Wung and Chen (36) who made heat transfer and temperature profile measurements on air-filled cavities of high aspect ratio and with $1.1x10^3<Ra<5x10^6$. In Figure 6.6 data taken from reference (36) are reproduced. Values of Gr and A for which Yin, Wung and Chen reported their measurements are shown and ranges of Ra and A over which they (and others) felt the conduction, transition and boundary layer regimes occur are presented. Elder (41) proposed lines that mark the onset of "wavelike" motions $(Ra>8x10^8 Pr^{0.5} A^{-3})$ and turbulence $(Ra>10^{10} A^{-3})$ based on his experiments using water $(Pr \approx 7)$. The slope of these lines corresponds well to the upper limit for which Yin, Wung and Chen reported experimental data. (A line of slope=−1/3 has been superimposed on Figure 6.6.) Yin, Wung and Chen stated that temperature fluctations occurred for high Ra and that data were reported only for experiments in which no fluctations were measured. This suggests that a steady, laminar flow existed over the ranges of Ra and A for which data where reported. At $A=40$, for example, a steady, laminar flow persists for Ra well in excess of 10^4. The data shown in Figure 6.6 also support the idea that turbulence commences at lower values of Ra for cavities with larger aspect ratios.

The analysis of the laminar natural convection in a vertical slot requires more than the consideration of conduction, transition and boundary layer flows. In 1965 Elder (40) reported on visualization experiments, using paraffin and silicone oil, in which he detected a steady secondary flow. This secondary flow consisted of a regular "cats-eye" pattern of transverse, stationary cells within the core of the base flow - with the flow in each cell rotating in the same direction as the base flow. At certain values of Ra counter-rotating cells (tertiary cells) were found in the regions between the secondary cells. Cellular patterns have also been visualized by Vest and Arpaci (44) in air, Korpela (52) in air, Seki, Fukusako and Inaba (59) in transformer oil and glycerin, and Choi and Korpela (62) in air. Figure 6.7 shows a sketch of secondary and tertiary cells shown in reference (59).

The nature of flow in the conduction and boundary layer regimes has been studied (43-48,50-52,56,57,59,62) and attempts to predict the critical Rayleigh number, Ra_c, at which hydrodynamic instability causes the onset of secondary flow have been made. These predictions (for $Pr=0.71$) fall in a relatively narrow band, ranging from $Ra_c=5{,}595$ (Vest and Arpaci (44)) to $Ra_c=7{,}827$ (Unny (48)). Vest and Arpaci (44) made a visual measurement of $Ra_c=6{,}177 \pm 10\%$. Hollands and Konicek (50) used a calorimetric method to determine Ra_c and reported $Ra_c=7{,}810 \pm 362$.

The physical balances governing the behaviour of the fill gas can be represented by mathematical expressions for the conservation of mass, momentum and energy. Simplified expressions describing the laminar 2-D fill-gas flow usually assume that the fluid is Newtonian, compressibility effects and viscous dissipation can be neglected and that fluid properties can be taken as constant except in the formulation of the buoyancy term. Leonardi and Reizes (74) examined the assumption of constant fluid properties and demonstrated its validity for cases where the temperature variation is less than 10% of the mean (absolute) temperature.

A variety of authors (including (21,27-31,53,66-72,74-83)) present numerical solutions to the equations of motion for a fluid in a rectangular enclosure with differentially heated vertical walls and either ZHF or LTP horizontal boundary conditions. The large majority of these studies do not address the problem of high aspect ratio. Raithby and Wong (75) gave finite-difference predictions for heat transfer across vertical air layers with $2<A<80$ and $10^3<Ra<3\times10^5$. A comparision between their results and the experimental results of ElSherbiny, et al (16) is shown in Figure 6.8 (data taken from reference (75)). Raithby and Wong were able to collapse the results of all aspect ratios onto a single curve (solid line in Figure 6.8) by plotting Nu vs. Ra^*. (Ra^* is obtained by multiplying Ra by a factor that was a function of A only.) Using the method of Bergholz (57) they also calculated the values of Ra^* at which hydrodynamic instabilities are expected. Figure 6.8 shows that, for each set of experimental data at a specific value of A, the predicted rate of heat transfer closely corresponds to the measured rate of heat transfer up to the critical value of Ra (or Ra^*) at which hydrodynamic instability is predicted and at which the onset of secondary flow is expected. This difference between the measured and predicted results can readily be explained because secondary cells were not predicted by the analysis. Raithby and Wong suggested that the secondary and tertiary flows might have been resolved if a finer grid had been used.

Several numerical models have been able to resolve secondary cells in the vertical slot (76,79-83). Some authors point out that the onset of secondary instabilities can be delayed by the false diffusion that results from numerical upwinding schemes. Each model capable of generating cells includes special provisions to accurately calculate the convection terms in the flow. Most models are based on high order discretization schemes (76,80,83,84). Others claimed to have avoided false diffusion by using a central difference discretization scheme and uniform grids (81,82).

Noteworthy results were produced by Lee and Korpela who (76) modeled laminar air flow in a vertical slot for $3.5x10^3<Ra< 1.75x10^5$. The results of this simulation included the onset of secondary cells at Ra_c between $7x10^3$ and $7.7x10^3$ with A=20. They pointed out that cells are not expected to develop for A<12. Streamline results for various values of Ra are shown in Figure 6.9. Lee and Korpela also compared their predicted values of Nu with the experimental results of ElSherbiny, Raithby and Hollands. This comparison is shown in Figure 6.10 (data taken from reference (76)). In this case, the predicted heat transfer rates were in close agreement with the measured heat transfer rates to appreciably higher values of Ra than was the case with the predictions of Raithby and Wong. For instance, at A=40 and $Ra=2x10^4$, the predictions of Lee and Korpela agree with experiment to within 10% while the predictions of Raithby and Wong show a discrepancy of 10% by $Ra^*=247$ ($Ra=1.2x10^4$). The improved agreement with measurement was attributed directly to their ability to resolve the secondary cells. The results of Lee and Korpela consistently underpredict the measured values of Nu at higher values of Ra ($Ra>1.2x10^4$ for A=40). This may be a result of the failure of their method to resolve a tertiary fluid flow or it is possible that $Ra=1.2x10^4$ marks the onset of turbulence for A=40.

Ramaman and Korpela (80) added to the work of Lee and Korpela (76) by using multi-grid methods to accelerate solution convergence.

Korpela, Lee and Drummond (79) showed that the results of Bergholz (57) could be simplified to predict the critical value of Grashof, Gr_c, at which the onset of secondary cells takes place from the conduction regime:

$$Gr_c = (1 + 5/A)/1.25x10^{-4} \tag{6.1}$$

In a window cavity, where A is very large, Equation 6.1 predicts the onset of secondary cells at Gr=8,000 or Ra=5,700.

Chait and Korpela performed a 3-D stability analysis for an infinitely tall and infinitely wide vertical slot (84). They warn that 3-D instabilities exist at values of Gr as small as 8,560. However, such 3-D flows have never been visualized experimentally in finite cavities and are not apparent in calorimetric experimental results.

Many experimental and numerical studies provide information regarding the local rate of convective heat transfer. Data taken from the numerical solution of Korpela, Lee and Drummond (79) are plotted in Figure 6.11. Figure 6.11 shows the local Nusselt number (denoted Nu(y)) as a function of the distance from the bottom of the cavity, y. Nu(y) is based on the heat flux at the warm vertical wall. The three curves shown correspond to three values of Ra. When Ra is sufficiently small (see Ra=3,550) the flow is in the conduction regime and Nu(y)=1 except at the ends of the cavity. At higher values of Ra the rate of heat transfer increases in most regions and the wave like nature of the curves in the middle portion of the cavity indicates that secondary cells are present.

Discussion

A simplified version of Figure 6.6 is given in Figure 6.12. The lines suggested by Yin, Wung and Chen (36) to delimit the laminar flow regimes are shown along with the line of slope=-1/3 below which steady flow is expected. The region of aspect ratio applicable to windows ($A>40$) is marked and the line representing the onset of secondary cells, given by Equation 6.1, is also shown. It can be seen that the the character of the convective flow in a window cavity is likely to move from conduction directly into secondary or turbulent flow as Ra increases. It is unlikely that either laminar transition or laminar boundary layer flows will exist.

The range of Ra expected in window cavities can be determined by 1-D simulation. Under the ASHRAE winter design condition, calculations (using VISION) show that $Ra=6.6x10^3$ ($Gr=9.3x10^3$) for a conventional double glazed window (1/2 inch pane spacing) and $Ra=8.3x10^3$ ($Gr=1.2x10^4$) for the same window with a soft low-e coating. Corresponding values of Ra for similar windows with argon fill gas are about 25% higher. If krypton is used in place of air with the same pane spacing Ra will be higher by a factor of about 4.5. When air or argon fill gas is used with 1/2 inch edge-seals the motion of the fill gas will be laminar and free of secondary cells under most conditions. In contrast, when krypton is used narrower gaps (smaller ℓ_x) and/or glazing systems with more glazings (smaller ΔT across each gap) are needed in order to reduce Ra to the point where turbulence can be avoided.

2-D NUMERICAL MODEL FOR NATURAL CONVECTION

Formulation

The physical balances present in the laminar 2-D fill-gas flow can be described mathematically by assuming that the fluid is Newtonian, compressibility effects and viscous dissipation can be neglected and that fluid properties can be taken as constant except in the formulation of the buoyancy term. The differential equations describing the conservation of energy, horizontal and vertical momentum and mass, respectively, at any point in the flow, are:

$$\rho C_p\left\{\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}\right\}=k\left\{\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right\}+S \qquad (6.2)$$

$$\rho\left\{\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}\right\}=\mu\left\{\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right\}-\frac{\partial P}{\partial x} \qquad (6.3)$$

$$\rho\left\{\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}\right\}=\mu\left\{\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right\}-\frac{\partial P}{\partial y}+\rho g\beta(T-T_m) \qquad (6.4)$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \qquad (6.5)$$

These coupled equations are elliptic in both the x and y dimensions. The boundary conditions are as shown in Figure 6.3:

$T = T_c$ at $x=0$ (6.6)

$T = T_h$ at $x=\ell_x$ (6.7)

$\frac{\partial T}{\partial y} = 0$ at $y=0$ and $y=\ell_y$ (i.e., ZHF) (6.8)

$u = v = 0$ at $y=0$, $y=\ell_y$, $x=0$ and $x=\ell_x$ (6.9)

A model has been formulated in order to simulate the problem shown in Figure 6.3. The analysis provides a numerical solution to the discretized versions of equations 6.2 through 6.5, applied in their integral form over the boundaries of each control volume, in an array of rectangular finite control volumes covering the problem domain.

The expected presence of secondary cells precludes the use of a non-uniform grid. Interesting detail can be expected everywhere in the flow so the solution algorithm divides the problem domain using uniform grid spacings in both the x and y directions. Figure 6.13 shows the staggered grid formulation that has been implemented (85). Note that the x-direction index, i, begins at i=ib and ends at i=ie. Similarly, the y-direction index runs from j=jb to j=je.

The formulation of the discretized equations is conservative with the time dependent terms retained in order to apply the method of the "false transient" to arrive at a steady-state solution. The algorithm used to solve the four sets of coupled equations, called SIMPLEC (86), is an extension of the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) which is described in detail by Patankar (87)[†]. An Alternating Direction Implicit (ADI) solver has been employed to solve each individual set of equations.

The numerical model uses either the central difference scheme (CDS) or an upwind difference scheme (UDS) to calculate the convective and diffusive fluxes of energy and momentum at each control volume face. The calculation of T, u or v (or their derivatives) is simplified by the use of a uniform grid. The value and spatial gradient of an entity, say ϕ, (ϕ can be T, u or v depending on whether flux of energy, x-momentum or y-momentum is being examined.) at the east face of a control volume is calculated as shown in Equations 6.10 and 6.11. The related geometry and nomenclature are shown in Figure 6.14.

$$\phi_e = (0.5 + \alpha)\phi_{i,j} + (0.5 - \alpha)\phi_{i+1,j} \tag{6.10}$$

$$\left.\frac{\partial\phi}{\partial x}\right|_e = \beta(\phi_{i+1,j} - \phi_{i,j})/\Delta x \tag{6.11}$$

Various relations (yielding results of comparable accuracy) are available for calculating α and β. The following power law upwinding scheme (88) has been chosen to calculate α and β as a functions of the grid Peclet number, Pe.

$$\alpha = Pe^2/(10 + 2Pe^2) \tag{6.12}$$

$$\beta = (1 + 0.005Pe^2)/(1 + 0.05Pe^2) \tag{6.13}$$

where,

$$Pe = \rho u_{i,j} C_p \Delta x/k \tag{6.14}$$

Similar calculations are carried out for the north control volume faces by substituting $v_{i,j}$, Δy and $\phi_{i,j+1}$ for $u_{i,j}$, Δx and $\phi_{i+1,j}$.

[†] Reference (87) is the best source of general information about SIMPLE. It deals with almost all aspects of the numerical model described here.

When high values of Pe occur the flux at a control volume face is dominated by convection so $\alpha \to 0.5$ and $\beta \to 0.1$. In this case, $\phi \to \phi_{i,j}$ and $\partial\phi/\partial x$ becomes small. When Pe is small (say $|Pe|<2$) α and β approach 0 and 1, respectively, and the upwinding scheme, given by Equations 6.10 to 6.14, approaches the central difference scheme. The central difference scheme can be imposed manually by setting $\alpha=0$ and $\beta=1$.

The central difference scheme and the upwind scheme are classified as second order accurate and first order accurate, respectively (87). Some researchers who have successfully modeled secondary cells in natural convection have found it necessary to use higher order schemes (76,79,80,83). For example, the model of Lee and Korpela (76) incorporates a fourth order scheme that requires a thirteen point computational molecule. Other models (81,82) have resolved secondary cells with CDS formulations. The numerical model used in the current study uses the CDS formulation when secondary cells are expected and UDS is used when cells are not expected.

The current model differs from SIMPLEC only in that the Additive Correction Method (ACM) presented by Hutchinson and Raithby (89) has been used to accelerate convergence when solving the pressure correction (p-prime) portion of the SIMPLEC procedure. The ACM algorithm is a multigrid method that is useful when individual control volumes are oblong. The control volumes must be much taller than they are wide in the analysis of the vertical slot in order to limit computer storage and run time. The ACM applies additive pressure corrections to blocks of control volumes by solving the p-prime set of equations for a course grid where each control volume consists of a single horizontal row of control volumes from the finer grid. The control volumes in the course grid are also oblong but their orientation is horizontal rather than vertical allowing for pressure correction information to be moved more rapidly in the vertical direction.

The ACM method does not alter the formulation of the governing equation nor does it alter its solution. It is used solely to speed the convergence of the iterative solver. Detailed information can found in references (80) and (89).

Grid Sizing

Throughout the current study the grid aspect ratio has been set equal to 5 (i.e., $A_{grid}=(\Delta y/\Delta x)=5$) as a compromise to limit computer requirements while avoiding an extreme imbalance in the areas of the vertical and horizontal faces of the finite control volumes. In their studies Raithby and Wong (75) used less than 2,000 grid points at A=40 (12x160) and Lee and Korpela (76) used slightly more than 2,000 (17x129) at A=20. Ramanan and Korpela (80) used a 32x256 grid (A>17.5), Korpela, Lee and Drummond (79) used a 17x65 grid (10<A<20) and de Vahl Davis and Jones (81) used a 11x41 grid (A=20). Grid aspect ratios used in these studies ranged from 2.5 to 5.4. Lauriat and Desrayaud set the number of vertical divisions equal to ten times the aspect ratio and used up to 31 horizontal divisions (10<A<20).

In order to test for grid dependence in the calculated results, simulations were completed with A=40 and Ra=8,239. The computed Nusselt number, Nu, (i.e., total heat transfer over the height of the cavity) did not differ in the third significant digit between the 25x200 and 30x240 grids. Nu changed by about 1% when the grid was reduced to 20x160. Secondary cells were expected (Ra_c=6,413) and 15 cells were modeled in each of the three simulations. All subsequent runs were carried out using 25 horizontal divisions.

Perturbation

Two previous studies have been mentioned where secondary cells were modeled using the CDS formulation. Cells were reported only for Ra as low as 20,000 in one case (81) and as low as about 22,000 in the other case (82). Both simulations used A=20 so secondary cells might have been expected for Ra>7,150 (see Equation 6.1). In the present study cells arose naturally from the numerical simulation at Ra=16,479 and A=40 using the CDS formulation. However, knowing that two or more solutions might exist at still lower Ra, it was thought that the simulation might select a solution with secondary cells if it were given a "push". It has been discovered that secondary cells can be modeled with Ra approaching the theoretical critical value by perturbing the flow.

The procedure used to generate secondary cells consists of several steps. First, the solution for a unicellular base flow is established using the upwind scheme. This can be done using relatively large time steps - say 50 times as great as the maximimum time step possible for stability in an explicit formulation. Second, the solution scheme is switched to CDS and the time step is reduced to the maximum explicit time step. Third, cells are introduced by summing the existing velocity field with a "perturbation" velocity field which resembles secondary cells alone. Finally, the SIMPLEC iteration process is allowed to continue to convergence. This step-wise procedure provides a useful way to directly examine the differences that secondary cells create.

The velocity perturbation field was formulated on the basis of information given by Lee and Korpela (76.) The height of a single secondary cell, ℓ_c, (or the distance between the centres of adjacent cells) can be expressed in terms of a wavenumber, α_c, according to Equation 6.15.

$$\ell_c = \ell_x(2\pi/\alpha_c) \tag{6.15}$$

α_c was calculated as a function of Gr by interpolating between the wavenumbers observed in the simulation results reported in reference (76). These are:

$\alpha_c = 2.82$ at Gr = 11,000
$\alpha_c = 2.50$ at Gr = 15,000
$\alpha_c = 2.41$ at Gr = 20,000
$\alpha_c = 2.33$ at Gr = 25,000

These data show that the secondary cells lengthen as Gr is increased; the increasing cell spacing can be seen in Figure 6.9. The theoretical wave number expected for cells at $Ra=Ra_c$ and $A \rightarrow \infty$ is $\alpha_c=2.8$ (57). When Gr was less than 11,000, α_c was set equal to 2.82.

The number of cells, n_c, is calculated using Equation 6.16 which is also based on data presented in reference (76).

$$n_c = \mathrm{int}[(A-10)/(2\pi/\alpha_c)] + 2 \qquad (6.16)$$

Once the number and size of secondary cells had been determined the perturbation velocity components, u_c and v_c, were calculated in terms of the perturbation stream function, Ψ_c, according to the convention:

$$u_c = -\,\partial\Psi_c/\partial y \qquad (6.17)$$

$$v_c = \partial\Psi_c/\partial x \qquad (6.18)$$

The perturbation stream function used for all simulations is given by the following equations.

$$\Psi_c = X(x)\cdot Y(y) \qquad (6.19)$$

where,

$$X(x) = \left[1 + \cos\{2\pi((x/\ell_x)-0.5)\}\right]/2 \qquad (6.20)$$

$$Y(y) = -\left[1 - \cos\{\alpha_c A((y/\ell_y)-0.5) + n_{o/e}\pi\}\right] \qquad (6.21)$$

$n_{o/e}=1$ if n_c is odd, $n_{o/e}=0$ if n_c is even

Equation 6.20 applies over the range:

$$(\ell_y - n_c\ell_c)/2 < y < (\ell_y + n_c\ell_c)/2$$

Otherwise, near the ends of the cavity,

$$Y(y) = 0 \qquad (6.22)$$

The components of the perturbation velocity, u_c and v_c conserve mass by virtue of their stream function formulation. The stream function, Ψ_c, has been chosen so that u_c and v_c also comply with the zero slip boundary condition set out in Equation 6.9.

The perturbation, Ψ_c, was applied over the solution grid and scaled such that the maximum value of u_c would be equal to the maximum value of u known to exist in the unicellular base flow, u_{max}. Using the discretized counterparts of Equations 6.17 and 6.18, the perturbation velocity components at the east and north faces of any control volume, u_c and v_c, respectively, are:

$$u_c = 2u_{max}(\Psi^{se}-\Psi^{ne})\cdot(je-jb+1)/(\alpha_c A) \tag{6.23}$$

$$v_c = 2u_{max}(\Psi^{ne}-\Psi^{nw})\cdot(ie-ib+1)/\alpha_c \tag{6.24}$$

The variables Ψ^{se}, Ψ^{ne} and Ψ^{nw} represent Ψ_c evaluated at the south-east, north-east and north-west corners of the control volume, respectively.

Solution Procedure

The steps followed to arrive at the numerical solution of the vertical slot natural convection problem are shown Figure 6.15. The procedure marches through time using the distorted transient formulation. The cycle of formulating coefficients and solving for each of T, u, v and the P-prime pressure correction, P', corresponds directly to the SIMPLEC algorithm. This cycle is called the coefficient update loop. The numerical solution of each of these variables was carried out using an ADI solver and ACM block correction was applied to speed the solution for P'.

The ADI solver is used to solve for a set of simultaneous equations of the form shown in Equation 6.25. Given a set of "a" and "b" coefficients for each control volume the solver returns the solution for $\phi_{i,j}$.

$$a_p\phi_{i,j} = a_e\phi_{i+1,j} + a_w\phi_{i-1,j} + a_n\phi_{i,j+1} + a_s\phi_{i,j-1} + b_p \tag{6.25}$$

Operation of the ADI solver was terminated either when it had executed a specified maximum number of passes over the problem domain or when it had solved $\phi_{i,j}$ to within a specified tolerence. The degree of solution convergence was determined by calculating the residual in the ϕ solution, R_ϕ, given by the root mean square (RMS) average of the residuals at each control volume, $R_{i,j}$, in the problem domain. $R_{i,j}$ is the difference between the two sides of Equation 6.25.

$$R_{i,j} = a_e\phi_{i+1,j} + a_w\phi_{i-1,j} + a_n\phi_{i,j+1} + a_s\phi_{i,j-1} + b_p - a_p\phi_{i,j} \quad (6.26)$$

The ideal goal in executing the solver is to reduce all $R_{i,j}$ to zero. The ADI solver was terminated if R_ϕ was reduced by a factor of 10^4. Otherwise, the number of ADI double-passes was limited to about 8, 4 and 5 while solving for T, u and v, respectively. The P' solution was allowed to run for 7 iterations following the application of ACM block correction. The combination of ACM correction plus 7 ADI iterations was repeated as many as 7 times – making for a possible total of 49 ADI iterations in solving for P'.

The desired residual reduction factor of 10^4 was readily reached for all variables early in the solution procedure with the P' solution (which always requires the majority of the CPU time) usually needing only 14 ADI passes. During coefficient loop passes late in the solution procedure there was little improvement to be made in the values of T, u, v and P' so a smaller amount of residual reduction was available. Generally, the residual reduction was between 10^3 and 10^2 for all variables, with P' taking 49 ADI passes, by the time convergence at the level of the coefficient loop was achieved.

The P' residual can be interpreted as the RMS average mass divergence (i.e., average amount of mass generated per control volume). The application of the P' solution in correcting the flow field consistently reduced the average mass divergence to about $0.5\text{x}10^{-11}$ Kg/sec. This corresponds to an average error in velocity at control volume faces of approximately $7\text{x}10^{-10}$ m/sec.

Convergence of the coefficient loop was determined by examining the rate of heat transfer (or Nusselt number) across various vertical planes in the vertical cavity.

Calculation of Nusselt Number

After a solution is found for the temperature and velocity fields the heat transfer can be quantified. It is customary to express the rate of heat transfer across the cavity in terms of a Nusselt number.

Since the top and bottom surfaces of the problem domain are adiabatic the heat transfer through any two vertical planes within the cavity must be equal. The Nusselt number, Nu, based on the average horizontal heat flux from the hot wall to the cold wall can be calculated by summing the conductive and convective components of heat transfer from the bottom to the top of any vertical cut:

$$\mathrm{Nu} = \int_{y=0}^{\ell_y} \{ k(\partial T/\partial x) - \rho C_p u(T - T_{ref}) \} \, dy/q_o \tag{6.27}$$

where, T_{ref} is a constant reference temperature

and q_o is a reference heat flux used to render Nu dimensionless and is chosen, consistent with convention, to be the heat flux expected if the gas layer were stagnant.

$$q_o = (k/\ell_x)(T_h - T_c) \tag{6.28}$$

Equation 6.27 can be simplified by splitting the convection term into two parts and noting that the net amount of mass crossing any vertical plane must be zero.

$$\int_{y=0}^{\ell_y} \rho C_p u T_{ref} \, dy = C_p T_{ref} \int_{y=0}^{\ell_y} \rho u \, dy = 0 \tag{6.29}$$

So it is unnecessary to specify a value for T_{ref} in order to calculate Nu.

$$\mathrm{Nu} = \int_{y=0}^{\ell_y} \{ k(\partial T/\partial x) - \rho C_p u T \} \, dy/q_o \tag{6.30}$$

The local Nusselt number, Nu(y) can be evaluated at either wall where u=v=0.

$$\mathrm{Nu}(y) = k(\partial T/\partial x)/q_o \qquad (\text{at } x=0 \text{ or } x=\ell_x) \tag{6.31}$$

It is apparent, because of the symmetry that exists, that Nu(y) at either of the vertical walls is equal to Nu(ℓ_y–y) at the opposite wall.

The discretized versions of Equations 6.30 and 6.31 have been used to calculate Nusselt numbers for the numerical simulation. It can be shown that Nu, calculated at a vertical plane running coincident with the right hand side of the i^{th} column of control volumes, is given by:

$$\mathrm{Nu} = \frac{n_x}{n_y} \sum_{j=jb}^{je} \{ \beta(T_{i+1,j}-T_{i,j}) - \mathrm{Pe}[(0.5+\alpha)T_{i,j} + (0.5-\alpha)T_{i+1,j}]\}/(T_h - T_c) \qquad (6.32)$$

where, n_x = ie–ib+1 = number of columns of control volumes
n_y = je–jb+1 = number of rows of control volumes

and the local values of α, β and Pe are given by Equations 6.12 to 6.14 if UDS is being used or α=0 and β=1 if CDS is being used.

The value of Nu at the hot vertical wall (where u=0, so Pe=α=0 and β=1) is:

$$\mathrm{Nu} = \frac{2n_x}{n_y} \sum_{j=jb}^{je} (T_h-T_{ie,j}) /(T_h - T_c) \qquad (6.33)$$

The local Nusselt number at the hot wall is simply:

$$\mathrm{Nu}(y) = 2n_x(T_h - T_{ie,j})/(T_h - T_c) \qquad (6.34)$$

Test for Convergence of the Coefficient Loop

Convergence of the solutions for T, u and v at the coefficient loop level was tested by calculating Nu at each vertical control volume division. The problem domain was divided horizontally into 25 columns of control volumes so it was possible to calculate Nu by applying Equation 6.32 in 26 different places. While making these calculations, the maximum, minimum and average values of the calculated Nusselt numbers, Nu_{max}, Nu_{min} and Nu_{avg}, respectively were noted.

The level of convergence was tested by calculating the percent difference between Nu_{max} and Nu_{min}, denoted $\Delta Nu_{\%}$.

$$\Delta Nu_{\%} = ((Nu_{max} - Nu_{min})/Nu_{avg}) \times 100\% \tag{6.35}$$

Simulations were run well beyond the number of iterations required for Nu_{max} and Nu_{min} to be equal and to remain unchanged in the third significant decimal place. Generally, these conditions were met if $\Delta Nu_{\%}$ was reduced to (or near) 0.1%.

The $\Delta Nu_{\%}<0.1\%$ condition could not be used as the sole criterion for convergence. When simulations were performed at higher values of Ra (Ra = 13,732 and 16,479 at A=40) $\Delta Nu_{\%}$ fell below 0.1% within 100 iterations of the coefficient loop following the perturbation of secondary cells but rose and fell again in a decaying cyclic fashion until after about 400 iterations it finally took on a steady cyclic pattern but always remained below 0.5%. Similar difficulties in obtaining convergence were noted by Raithby and Wong (72). In order to guard against early termination, the program was run for a fixed number of iterations and the results were examined manually. In cases where more iterations were required it was possible to restart the simulation where it had left off.

Results

In order to test the numerical model, Nusselt number results were compared against two sets of measured results (16,87) taken using the UWNCA guarded heater plate apparatus. Simulations were performed with Ra as high as 16,479. The aspect ratio was fixed at A=40. This was the value of A closest to the aspect ratio at which subsequent SGU simulations were expected to be performed (A≈ 40 to 50) and for which results were included in both sets of experimental data.

Figure 6.16 shows Nu plotted as a function of Ra and includes the simulation results of Raithby and Wong (72), Lee and Korpela (76), the present study (with and without cells) plus measured values of Nu from ElSherbiny, Raithby and Hollands (16) and Shewen (90). A mix of LTP and ZHF results is shown in Figure 6.16. Note that Lauriat and Desrayaud (82) pointed out, on the basis of their own investigation, that the influence of LTP or ZHF boundary conditions (on Nu) is small for A>10. The measured results are marked by a band which indicates the reported uncertainty levels. The straight lines connecting pairs of simulation data are intended to distinguish between sets of data and are not included as a suggested functional dependence. The simulation results from the present study are also listed in Table 6.1.

Several observations can be made regarding the data shown in Figure 6.16. The measured data of ElSherbiny, Raithby and Hollands show the same trend as the measured data of Shewen but the Nu values given by Shewen consistently lie about 4% higher than the data of Elsherbiny et al. The reason for this discrepancy is not apparent. Shewen points out that the data of ElSherbiny et al for A=40 were not consistent with the data from the same source for A=80 and A=110. The data for A=40 depart from the Nu=1 condition at a higher critical Rayleigh number than expected – suggesting that the A=40 data of ElSherbiny may be slightly low.

Table 6.1: Nu vs. Ra Simulation Results, A=40

Run #	Ra	Cells Perturbed	Cells Kept	Nu without cells	Nu with cells
1	2,746	-	-	1.05	-
2	5,493	-	-	1.11	-
3	6,413	-	-	1.12	-
5	6,800	15	15	1.13	1.15
7	8,239	15	15	1.16	1.21
9	10,102	14	14	1.20	1.28
11	10,986	13	13	1.22	1.31
13	13,732	13	13	1.27	1.38
15	16,479	13	11	1.33	1.44

Values of Nu calculated without perturbation of secondary cells agree with the simulation of Raithby and Wong to within about 1%. These two sets of calculated results follow the same trend that moves progressively further from the measured data for $Ra>Ra_c$ (Ra_c=6,413 according to Equation 6.1) where the onset of secondary cells is expected. At Ra≈10,000 the value of Nu for flows without cells is about 6% and 10% lower than the data of Elsherbiny et al and Shewen, respectively. The calculated Nu data agree with the measured data of Shewen to within about 1% and to within 5% or less with the measured data of ElSherbiny et al for $Ra<Ra_c$.

Values of Nu calculated with cells follow closely the results of Lee and Korpela (also with cells) with the discrepancy being 2% or less. In the range of $Ra_c<Ra<10{,}000$ both of these sets of simulation results closely follow the measured data of Shewen. At Ra>10,000 both sets of results begin to depart from the trend of the measured data. However, the calculated value of Nu does not fall below the measured data of ElSherbiny et al until about Ra=14,000. Clearly, the simulation is not fully modeling all of the physical mechanisms of heat transfer for Ra>10,000. Fortunately, the range of Ra of interest for the simulation of glazing systems does not extend appreciably beyond Ra=10,000.

In addition to reporting Nu, ElSherbiny (91) and Shewen (90) provide measured average values of Nusselt for each of the metering areas over which heat transfer was measured. ElSherbiny measured heat transfer over three heater plates (See Figures 4.1 and 4.3) and Shewen measured heat transfer over five equal areas from the bottom to the top of the warm wall of the cavity. In order to make more detailed comparisons between calculation and measurement, the corresponding average values of the Nusselt number were calculated from simulation results by integrating Nu(y) along the hot wall within the appropriate limits of y.

The two sets of measured heat transfer results include data for almost equal values of Ra twice within the range of interest. Data were measured by ElSherbiny at Ra=6,936 and 10,072 and Shewen presents data for Ra=6,410 and 10,120. Two special simulation runs were executed to examine local heat transfer rates at Ra= 6,800 and 10,100. The results of these simulations are shown, in a variety of forms, in Figures 6.17 through 6.21.

Figure 6.17 shows stream function plots generated by computer simulation for Ra=6,800 and 10,100. In each case secondary cells were generated – 15 cells at Ra=6,800 and 14 cells at Ra=10,100. Note that the horizontal scale has been expanded to show the detail in the flow.

Figure 6.18 shows the calculated Nu(y) vs. y at $x=\ell_x$ and for Ra=6,800. Also shown are the average values of Nusselt number calculated for the three areas corresponding to the measurements areas of ElSherbiny. Similar results are shown in Figure 6.19 for Ra=10,100 but in this case the locally averaged Nusselt results correspond to the five measurement areas of Shewen. Both plots of Nu(y) include wiggles that coincide with the location of the secondary cells shown in Figure 6.17. Comparisons between the calculated meter-area Nusselt numbers and the data of ElSherbiny and Shewen are shown in Tables 6.2 and 6.3. In order to be complete results were also generated using the LTP boundary condition (for this comparison only) and are included in Tables 6.2 and 6.3. Little difference can be found between the LTP and ZHF results. Agreement between calculation and measurement was excellent.

Figures 6.20 and 6.21 each show colour fringe plots of T, u and v at the bottom end of the cavity for Ra=6,800 and Ra=10,100, respectively. These plots were generated using PATRAN (a Finite Element Method (FEM) pre/post processor) to display the finite volume results on a DEC 4125 work station where they were photographed. In these fringe plots the x direction scale has not been expanded in relation to the y direction.

DISCUSSION

The Nusselt number data provided by the 2-D finite volume analysis of the vertical slot for A=40 demonstrate an impressive capability for the detailed simulation of fill gas flow and heat transfer. Within the range of interest ($Ra<10^4$) the average Nusselt number, Nu, was consistently calculated to within 5% agreement with measured values where the discrepancy between the two available sets of measured data was no better than 4%. The ability to model secondary cells contributes to the accuracy achieved for $Ra>Ra_c$ and allows the Nu vs. Ra curve to follow the correct trend up to $Ra=10^4$.

The simulation has also demonstrated an ability to precisely predict local Nusselt numbers for the metering areas used in the experiments of ElSherbiny and Shewen. Again, local Nusselt numbers were predicted generally to within 5% and in most instances to within 2%. These discrepancies are well within experimental uncertainty.

It is concluded that the 2-D finite volume analysis developed and demonstrated here realistically models the important physical mechanisms present in the vertical slot problem and that analysis results, such as those shown in Figures 6.20 and 6.21, can be used both quantitatively and qualitatively to better understand these mechanisms. This simulation method can readily be extended to provide a useful analysis of the fill gas cavity between the panes of a glazing system.

Table 6.2
Meter-Area Nusselt Numbers –
Calculation vs. Measurements of ElSherbiny, A=40

Meter Location	Nusselt No. ElSherbiny (Ra=6,936) LTP	Nusselt No. Calculated (Ra=6,800) ZHF/LTP	Difference (%)	Nusselt No. ElSherbiny (Ra=10,072) LTP	Nusselt No. Calculated (Ra=10,100) ZHF/LTP	Difference (%)
Top	0.92	0.92/0.92	0/0	0.97	0.99/1.00	2/3
Middle	1.03	1.03/1.02	0/1	1.16	1.14/1.14	2/2
Bottom	1.34	1.38/1.36	3/1	1.55	1.56/1.53	1/2
Average	1.10	1.15/1.12	5/2	1.23	1.28/1.25	4/2

Table 6.3
Meter-Area Nusselt Numbers –
Calculation vs. Measurements of Shewen, A=40

Meter Location	Nusselt No. Shewen (Ra=6,410) LTP	Nusselt No. Calculated (Ra=6,800) ZHF/LTP	Difference (%)	Nusselt No. Shewen (Ra=10,120) LTP	Nusselt No. Calculated (Ra=10,100) ZHF/LTP	Difference (%)
Top	0.81	0.79/0.79	3/3	0.84	0.84/0.85	0/1
	1.02	1.03/1.03	1/1	1.18	1.13/1.14	4/3
Middle	1.02	1.02/1.02	0/0	1.18	1.14/1.14	3/3
	1.04	1.04/1.04	0/0	1.20	1.13/1.13	6/6
Bottom	1.72	1.82/1.74	6/1	2.05	2.10/2.00	2/2
Average	1.12	1.15/1.12	3/0	1.29	1.28/1.25	1/3

REFERENCES

(1) Solvason, K.R., "Pressures and Stresses in Sealed Double Glazing Units," National Research Council of Canada, Division of Building Research, Technical Paper No. 423., August, 1974.

(2) Wright, J.L., Sullivan, H.F., "Glazing System U-Value Measurement Using the a Guarded Heater Plate Apparatus," ASHRAE Window U-Value Measurements Symposium, Ottawa, June, 1988, ASHRAE Transactions, Vol. 94, Pt. 2.

(3) Frank, T., Mühlebach, H., "Coefficient de Transmission Thermique des Fenêtres," taken from an article in Ingénieurs et Architectes Suisse, 1987.

(4) IEA Energy Conservation in Buildings and Community Systems Programme, Annex XII, Windows and Fenestration, Step 1 - Building Regulations, Standards and Codes Concerning Thermal and Solar Performance of Windows; A Survey of Eight Countries, September, 1986.

(5) IEA Energy Conservation in Buildings and Community Systems Programme, Annex XII, Windows and Fenestration, Step 2 - Thermal Transmission Through Windows (Selected examples to illustrate the need for a more standardized approach).

(6) IEA Energy Conservation in Buildings and Community Systems Programme, Annex XII, Windows and Fenestration, Step 2 - Thermal and Solar Properties of Windows (Expert Guide).

(7) IEA Energy Conservation in Buildings and Community Systems Programme, Annex XII, Windows and Fenestration, Step 3 - Calculation of Seasonal Heat Loss and Gain Through Windows; A Comparison of Some Simplified Models, Sept., 1986.

(8) IEA Energy Conservation in Buildings and Community Systems Programme, Annex XII, Windows and Fenestration, Step 4 - Comparison of Six Simulation Codes DEROB DYWON PASSIM DOE-2.1C SERI-RES HELIOS1, Jan., 1987.

(9) IEA Energy Conservation in Buildings and Community Systems Programme, Annex XII, Windows and Fenestration, Step 5 - Windows and Space Heating Requirements (Parameter studies leading to a simplified calculation method).

(10) Curcija, D., Ambs, L.L., Goss, W.P., "A Comparison of European and North American Window U-Value Calculation Procedures," ASHRAE Transactions, Vol. 95, Pt. 1, presented at the ASHRAE annual winter meeting, Chicago, January, 1989.

(11) ASHRAE. 1989. ASHRAE Handbook – 1989 Fundamentals, Atlanta: American Society of Heating, Refrigerating and Air Conditioning Engineers, Inc.

(12) Jonsson, B., "Heat Transfer Through Windows During the Hours of Darkness with the Effect of Infiltration Ignored," Swedish Council for Building Research report, Document D13:1985, March, 1985.

(13) "KOBRU86 Computerprogram to Calculate Two Dimensional Steady State Heat Transfer in Objects, Described in a Rectangular Grid, Using the Energy Balance Technique. Alphanumerical Input, Alphanumerical and Graphical Output," Information Brochure, June, 1986.

(14) Carpenter, S.C., "The Effect of Frame Design on Window Heat Loss - Phase 1," report prepared by Enermodal Engineering Ltd. for Energy, Mines and Resources Canada, Sept., 1987.

(15) Carpenter, S.C., McGowan, A., "Calculating Window U-Values Including Frame Effects," Proceedings of the 14th Annual Conference of the Solar Energy Society of Canada Inc., Ottawa, June, 1988.

(16) ElSherbiny, S.M., Raithby, G.D., Hollands, K.G.T., "Heat Transfer by Natural Convection Across Vertical and Inclined Air Layers," Journal of Heat Transfer, Vol. 104, Feb., 1982.

(17) Jakob, M., "Free Heat Convection Through Enclosed Plane Gas Layers," Trans. Amer. Soc. Mech. Eng., Vol. 68, 1946, pp. 189-194.

(18) Peck, R.E., Fagan, W.S., Werlein, P.P., "Heat Transfer Through Gases at Low Pressures," Trans. ASME, Vol. 73, 1951, pp. 281-287.

(19) DeGraaf, J.G.A., Van Der Held, E.F.M., "The Relation Between the Heat Transfer and Convection Phenomena in Enclosed Plane Air Layers," Applied Science Research, Vol. 3, 1953, pp. 393-409.

(20) Batchelor, G.K., "Heat Transfer by Free Convection Across a Closed Cavity Between Vertical Boundaries at Different Temperatures," Quarterly of Applied Mathematics, Vol. 12, 1954, pp. 209-233.

(21) Poots, G., "Heat Transfer by Laminar Free Convection in Enclosed Plane Gas Layers," Quart. J. Mech. Appl. Math., Vol. 11, 1958, pp.257-267.

(22) Eckert, E.R.G., Carlson, W.O., "Natural Convection in an Air Layer Enclosed Between Two Vertical Plates with Different Temperatures," International Journal of Heat and Mass Transfer, Vol. 2, 1961, pp. 106-120.

(23) Emery, A.F., "The Effect of a Magnetic Field Upon the Free Convection of a Conducting Fluid," Journal of Heat Transfer, Series C, Vol. 85, 1963, pp. 119-124.

(24) Dropkin, D., Somerscales, E., "Heat Transfer by Natural Convection in Liquids Confined by Two Parallel Plates Which are Inclined at Various Angles with Respect to the Horizontal," Journal of Heat Transfer, Series C, Vol. 87., 1965, pp. 77-84.

(25) Emery, A., Chu, N.C., "Heat Transfer Across Vertical Layers," Journal of Heat Transfer, Series C, Vol. 87, 1965, pp. 110-116.

(26) Landis, F., Yanowitz, H., "Transient Natural Convection in a Narrow Vertical Cell," Transactions, International Heat Transfer Conference, 1966, pp. 139-151.

(27) Elder, J.W., "Numerical Experiments with Free Convection in a Vertical Slot," J. Fluid Mechanics, Vol. 24, 1966, pp. 823

(28) de Vahl Davis, G., "Laminar Natural Convection in an Enclosed Rectangular Cavity," International Journal of Heat and Mass Transfer, Vol. 11, 1968, pp. 1675-1693.

(29) MacGregor, R.K., Emery, A.F., "Free Convection Through Vertical Plane Layers - Moderate and High Prandtl Number Fluids," Journal of Heat Transfer, August, 1969, pp. 391-403.

(30) Newell, M.E., Schmidt, F.W., "Heat Transfer by Laminar Natural Convection within Rectangular Enclosures," Journal of Heat Transfer, Series C, Vol. 92, 1970, pp. 159-165.

(31) Thomas, R.W., "Natural Convection in Annular and Rectangular Cavities a Numerical Study," Proceedings of the International Heat Transfer Conference, Paris, 1970.

(32) Jannot, M., Mazeas, C., "Etude Experimentale de la Convection Naturelle dans des Cellules Rectangulaires, Verticales," International Journal of Heat and Mass Transfer, Vol. 16, 1973, pp. 81-100.

(33) Mynett, J.A., Duxbury, D., "Temperature Distributions Within Enclosed Plane Air Cells Associated With Heat Transfer by Natural Convection," Proceedings of the 1974 International Heat Transfer Conference, Tokyo, pp. 119-123.

(34) Berkovsky, B.M., Polevikov, V.K., "Numerical Study of Problems on High-Intensive Free Convection," Heat Transfer and Turbulent Buoyant Convection, Vols. 1 and 2, Eds. Spalding and Afgan, 1977, Hemisphere Publishing Corp., pp. 443-455.

(35) Raithby, G.D., Hollands, K.G.T., Unny, T., "Analysis of Heat Transfer by Natural Convection Across Vertical Fluid Layers," Journal of Heat Transfer, Vol. 99, 1977, pp. 287-293.

(36) Yin, S.H., Wung, T.Y., Chen, K., "Natural Convection in an Air Layer Enclosed Within Rectangular Cavities," International Journal of Heat and Mass Transfer, Vol. 21, 1978, pp. 307-315.

(37) Seki, N., Fukusako, S., Inaba, H., "Heat Transfer of Natural Convection in a Rectangular Cavity with Vertical Walls of Different Temperatures," Bulletin of the JSME, Vol. 21, No. 152, February, 1978.

(38) Schinkel, W.M.M., Hoogendoorn, C.J., "An Interferometric Study of the Local Heat Transfer by Natural Convection in Inclined Airfilled Enclosures," Proceedings of the 1978 International Heat Transfer Conference (Toronto), pp. 287-292.

(39) Randall, K.R., Mitchell, J.W., El-Wakil, M.M., "Natural Convection Heat Transfer Characteristics of Flat Plate Enclosures," Journal of Heat Transfer, Vol. 101, 1979, pp. 120-125.

(40) Elder, J.W., "Laminar Free Convection in a Vertical Slot," Journal of Fluid Mechanics, Vol. 23, Part 1, 1965, pp. 99-111.

(41) Elder, J.W., "Turbulent Free Convection in a Vertical Slot," Journal of Fluid Mechanics, Vol. 23, Part 1, 1965, pp. 77-98.

(42) Gill, A.E., "The Boundary-Layer Regime for Convection in a Rectangular Cavity," Journal of Fluid Mechanics, Vol. 26, Part 3, 1966, pp. 515-536.

(43) Birikh, R.V., Gershuni, G.Z., Zhukhovitskii, E.M., Rudadov, R.N., "Hydrodynamic and Thermal Instability of a Steady Convective Flow," Prikl. Mat. Mekh., Vol. 32, No. 2, 1968, pp. 256-263.

(44) Vest, C.M., Arpaci, V.S., "Stability of Natural Convection in a Vertical Slot," Journal of Fluid Mechanics, Vol. 36, Part 1, 1969, pp. 1-15.

(45) Gill, A.E., Davey, A., "Instabilities of a Buoyancy-Driven System," Journal of Fluid Mechanics, Vol. 35, Part 4, 1969, pp. 775-798.

(46) Gill, A.E., Kirkham, C.C., "A Note on the Stability of Convection in a Vertical Slot," Journal of Fluid Mechanics, Vol. 42, Part 1, 1970, pp. 125-127.

(47) Hart, J.E., "Stability of the Flow in a Differentially Heated Inclined Box," Journal of Fluid Mechanics, Vol. 47, Part 3, 1971, pp. 547-576.

(48) Unny, T.E., "Thermal Instability in Differentially Heated Inclined Fluid Layers," Journal of Applied Mechanics, March, 1972, pp. 41-46.

(49) Brooks, R.G., Probert, S.D., "Heat Transfer Between Parallel Walls: An Interferometric Investigation," Journal of Mechanical Engineering Science, Vol 14, No. 2, 1972, pp. 107-127.

(50) Hollands, K.G.T., Konicek, L., "Experimental Study of the Stability of Differentially Heated Inclined Air Layers," Int. Journal of Heat and Mass Transfer, Vol. 16, 1973, pp. 1467-1476.

(51) Korpela, S.A., Gözüm, D., Baxi, C., "On the Stability of the Conduction Regime of Natural Convection in a Vertical Slot," Int. Journal of Heat and Mass Transfer, Vol. 16, 1973, pp. 1683-1690.

(52) Korpela, S.A., "A Study on the Effect of Prandtl Number on the Stability of the Conduction Regime of Natural Convection in an Inclined Slot," Int. Journal of Heat and Mass Transfer, Vol. 17, 1974, pp. 215-222.

(53) Catton, E., Ayyaswamy, P.S., Clever, R.M., "Natural Convection Flow in a Finite, Rectangular Slot Arbitrarily Oriented with Respect to the Gravity Vector," Int. Journal of Heat and Mass Transfer, Vol. 17, 1974, pp. 473-484.

(54) Dixon, M., Probert, S.D., "Heat-Transfer Regimes in Vertical, Plane-Walled, Air-Filled Cavities," Int. Journal of Heat and Mass Transfer, Vol. 18., 1975, pp. 709-710.

(55) Dulnev, G.N., Zarichnyak, Y.P., Sharkov, A.V., "Free Convection at a Vertical Plate and in Closed Interlayer at Different Gas Pressures," Int. Journal of Heat and Mass Transfer, Vol. 18, 1975, pp. 213-218.

(56) Clever, R.M., Busse, F.H., "Instabilities of Longitudinal Convection Rolls in an Inclined Layer," Journal of Fluid Mechanics, Vol. 81, Part 1, 1977, pp. 107-127.

(57) Bergholz, R.F., "Instability of Steady Natural Convection in a Vertical Fluid Layer," Journal of Fluid Mechanics, Vol. 84, Part 4, 1978, pp. 743-768.

(58) Morrison, G.L., Tran, V.Q., "Laminar Flow Structure in Vertical Free Convective Cavities," Int. Journal of Heat and Mass Transfer, Vol. 21, 1978, pp. 203-213.

(59) Seki, N., Fukusako, S., Inaba, H., "Visual Observation of Natural Convective Flow in a Narrow Vertical Cavity," Journal of Fluid Mechanics, Vol. 84, Part 4, 1978, pp. 695-704.

(60) Catton, I., "Natural Convection in Enclosures," Proceedings of the International Heat Transfer Conference, 1978, Toronto, Canada.

(61) Bejan, A., "Note on Gill's Solution for Free Convection in a Vertical Enclosure," Journal of Fluid Mechanics, Vol. 90, Part 3, 1979, pp. 561-568.

(62) Choi, I.G., Korpela, S.A., "Stability of the Conduction Regime of Natural Convection in a Tall Vertical Annulus," Journal of Fluid Mechanics, Vol. 99, Part 4, 1980, pp. 725-738.

(63) Graebel, W.P., "The Influence of Prandtl Number on Free Convection in a Rectangular Cavity," Int. Journal of Heat and Mass Transfer, Vol. 24, 1981, pp. 125-131.

(64) ElSherbiny, S.M., Hollands, K.G.T., Raithby, G.D., "Effect of Thermal Boundary Conditions on Natural Convection in Vertical and Inclined Air Layers," Journal of Heat Transfer, Vol. 104, August, 1982, pp. 515-520.

(65) ElSherbiny, S.M., Hollands, K.G.T., Raithby, G.D., "Nusselt Number Distribution in Vertical and Inclined Air Layers," Transactions of the ASME, Vol 105, May, 1983, pp. 406-408.

(66) de Vahl Davis, G., "Laminar Natural Convection in an Enclosed Rectangular Cavity," Int. Journal of Heat and Mass Transfer, Vol. 11, 1968, pp. 1675-1693.

(67) Quon, C., "High Rayleigh Number Convection in an Enclosure - A Numerical Study," The Physics of Fluids, Vol. 15, No. 1, January, 1972, pp. 12-19.

(68) Mallinson, G.D., de Vahl Davis, G., "The Method of the False Transient for the Solution of Coupled Elliptic Equations," Journal of Computational Physics, Vol. 12, 1973, pp. 435-461.

(69) Spradley, L.W., Churchill, S.W., "Pressure and Buoyancy-Driven Thermal Convection in a Rectangular Enclosure," Journal of Fluid Mechanics, Vol. 70, Part 4, 1975, pp. 705-720.

(70) de Vahl Davis, G., Mallinson, G.D., "A Note on Natural Convection in a Vertical Slot," Journal of Fluid Mechanics, Vol. 72, Part 1, 1975, pp. 87-93.

(71) Mallinson, G.D., de Vahl Davis, G., "Three-Dimensional Natural Convection in a Box: A Numerical Study," Journal of Fluid Mechanics, Vol. 83, Part 1, 1977, pp. 1-31.

(72) Wong, H.H., Raithby, G.D., "Improved Finite-Difference Methods Based on a Critical Evaluation of the Approximation Errors," Numerical Heat Transfer, Vol. 2, 1979, pp. 139-163.

(73) Jones, I.P., "A Comparison Problem for Numerical Methods in Fluid Dynamics, The 'Double-Glazing' Problem," Proceedings of the International Heat Transfer Conference, 1979, pp. 338-348.

(74) Leonardi, E., Reizes, J.A., "Natural Convection in Compressible Fluids with Variable Properties," Proceedings of the International Heat Transfer Conference, 1979, pp. 297-306.

(75) Raithby, G.D., Wong, H.H., "Heat Transfer by Natural Convection Across Vertical Air Layers," Numerical Heat Transfer, Vol. 4, 1981, pp. 447-457.

(76) Lee, Y., Korpela, S.A., "Multicellular Natural Convection in a Vertical Slot," Journal of Fluid Mechanics, Vol. 126, 1983, pp. 91-121.

(77) de Vahl Davis, G., "Natural Convection of Air in a Square Cavity: A Bench Mark Numerical Solution," International Journal for Numerical Methods in Fluids, Vol. 3, 1983, pp. 249-264.

(78) ElSherbiny, S.M., Fath, H.E.S., Refai, G.A., "Numerical Analysis of Natural Convection in Vertical and Inclined Air Layers," Submitted for publication in the Journal of Heat and Fluid Flow, 1987.

(79) Korpela, S.A., Lee, Y., Drummond, J.E., "Heat Transfer Through a Double Pane Window," Journal of Heat Transfer, Vol. 104, pp.539-544, August, 1982.

(80) Ramanan, N, Korpela, S.A., "Multigrid Solution of Natural Convection in a Vertical Slot," Numerical Heat Transfer, Part A, vol. 15, pp. 323-339, 1989.

(81) de Vahl Davis, G., Jones, I.P., "The Effect of Vertical Temperature Gradients on Multi-Cellular Flows in High Aspect Ratio Cavities," proc. Conference on Liquid Metal Technology in Energy Production, Oxford, 9-13, 1984.

(82) Lauriat, G., Desrayaud, G., "Natural Convection in Air-Filled Cavities of High Aspect Ratio: Discrepancies Between Experimental and Theoretical Results," proc. National Heat Transfer Conference, ASME, Denver, Colorado, August 4-7, 1985.

(83) Roux, B., Grondin, J-C, Bontoux, P., "Natural Convection in Inclined Rectangular Cavities," proc. International Conference on Numerical Methods in Thermal Problems, Swansea, Eng.: Pineridge Press, 1979.

(84) Chait, A., Korpela, S.A., "The Secondary Flow and Its Stability for Natural Convection in a Tall Vertical Enclosure," Journal of Fluid Mechanics, vol. 200, pp. 189-216, 1989.

(85) Harlow, F.H., Welch, J.E., "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface," Physics of Fluids, Vol. 8, p. 2182, 1965.

(86) Vandoormaal, J.P., Raithby, G.D., "Enhancements of the SIMPLE Method," Numerical Heat Transfer, Vol. 7., pp. 147-163, 1984.

(87) Patankar, S.V., "Numerical Heat Transfer and Fluid Flow", Hemisphere Publishing Corp, 1980.

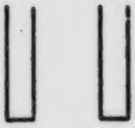
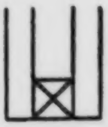
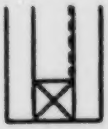
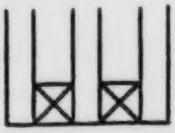
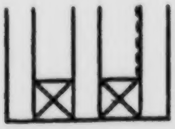
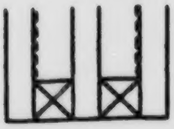
(88) Raithby, G.D., Galpin, P.F., VanDoormaal, J.P., "Prediction of Heat and Fluid Flow in Complex Geometries Using General Orthogonal Coordinates," Numerical Heat Transfer, Vol. 9, pp. 125-142, 1986.

(89) Hutchinson, B.R., Raithby, G.D., "A Multigrid Method Based on the Additive Correction Strategy," Numerical Heat Transfer, vol. 9, pp. 511-537, 1986.

(90) Shewen, E.C., "A Peltier-Effect Technique for Natural Convection Heat Flux Measurement Applied to the Rectangular Open Cavity," Ph.D. thesis, Department of Mechanical Engineering, University of Waterloo, (1986).

(91) ElSherbiny, S., "Heat Transfer by Natural Convection Across Vertical and Inclined Air Layers," Ph.D. thesis, Department of Mechanical Engineering, University of Waterloo, (1980).

(92) Wright, J.L., "Presentation of External Data Using the PATRAN FEM Pre/Post Processor," project report prepared for Prof. R.J. Pick, Department of Mechanical Engineering, University of Waterloo, January, 1989.

Glazing System	Linear Edge-Seal Conductance (W/m·K)
	0
	0.05 (0.04 - 0.06)
	0.06 (0.05 - 0.07)
	0.04 (0.03 - 0.05)
	0.05 (0.04 - 0.06)
	0.06 (0.05 - 0.07)

Dashed lines represent low emissivity coatings

Figure 6.1: Linear k-values Used to Estimate Edge-Seal Heat Loss (from reference (3))

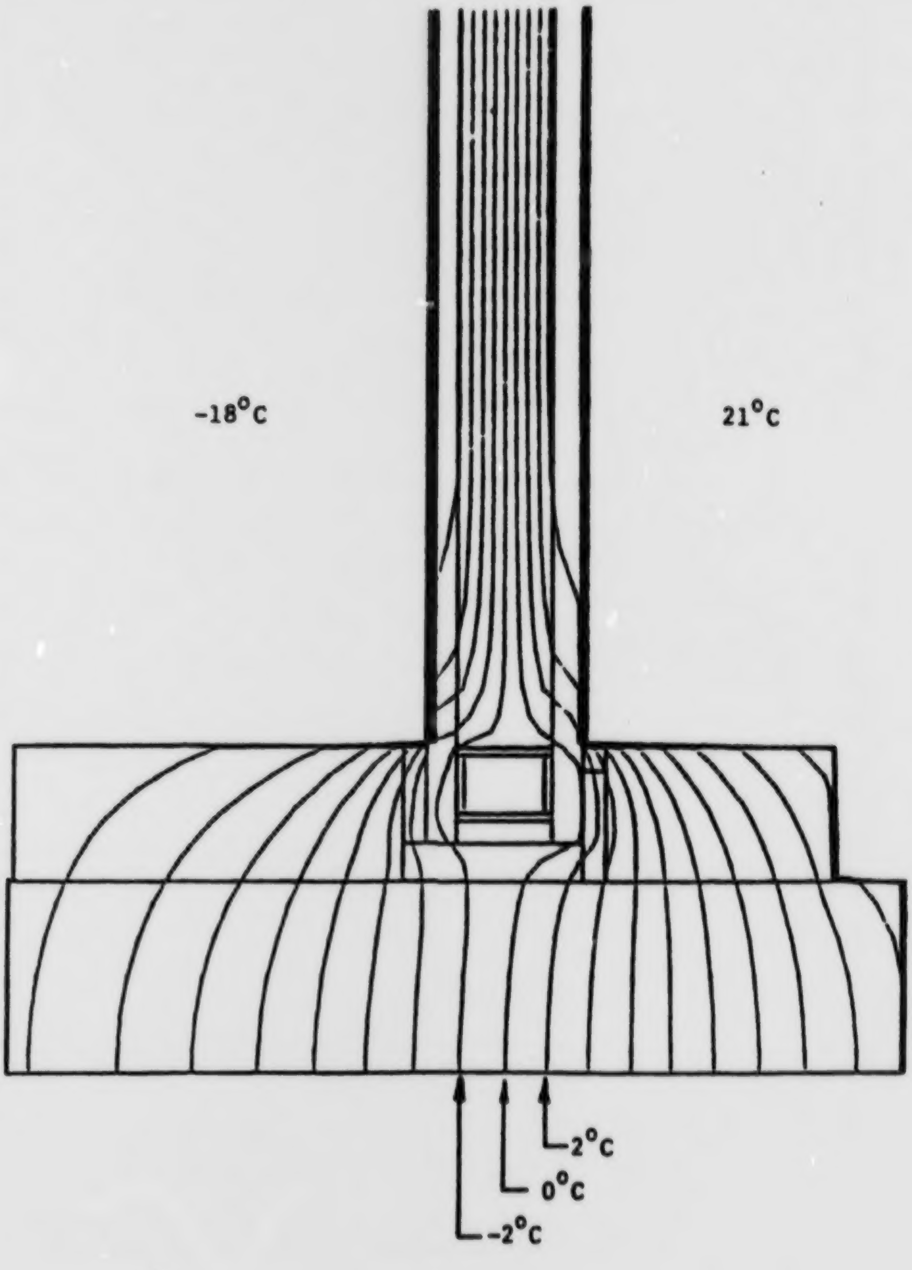

-18°C
21°C
2°C
0°C
-2°C

Figure 6.2: Plot Showing Isotherms: Standard Double Glazed Window, Wood Frame, Aluminum Spacer (from reference (14))

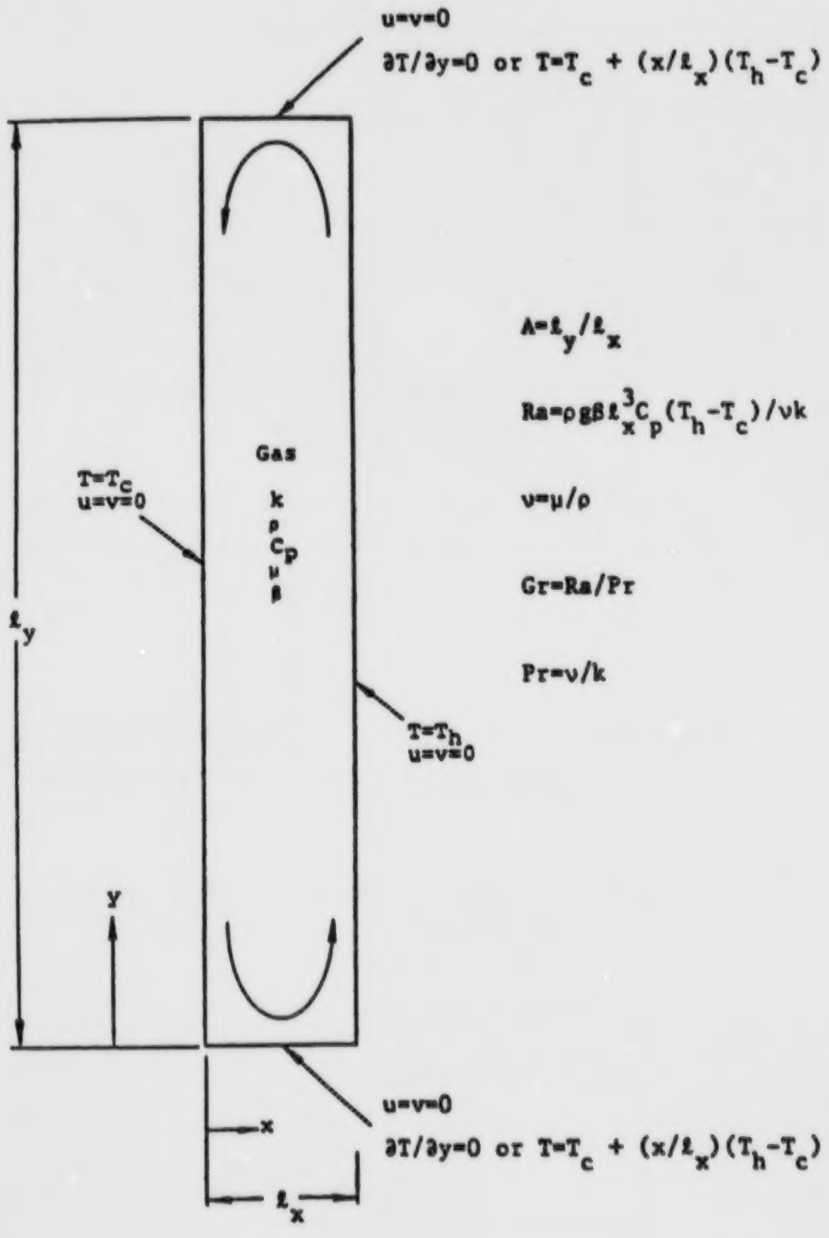

$u=v=0$
$\partial T/\partial y=0$ or $T=T_c + (x/\ell_x)(T_h-T_c)$
$A=\ell_y/\ell_x$
$Ra=\rho g\beta\ell_x^3C_p(T_h-T_c)/\nu k$
Gas
k
ρ
C_p
μ
β
$T=T_c$
$u=v=0$
$\nu=\mu/\rho$
$Gr=Ra/Pr$
ℓ_y
$Pr=\nu/k$
$T=T_h$
$u=v=0$
y
$u=v=0$
x
$\partial T/\partial y=0$ or $T=T_c + (x/\ell_x)(T_h-T_c)$
ℓ_x

Figure 6.3: Problem Domain for the Analysis of Natural Convection in a Vertical Slot

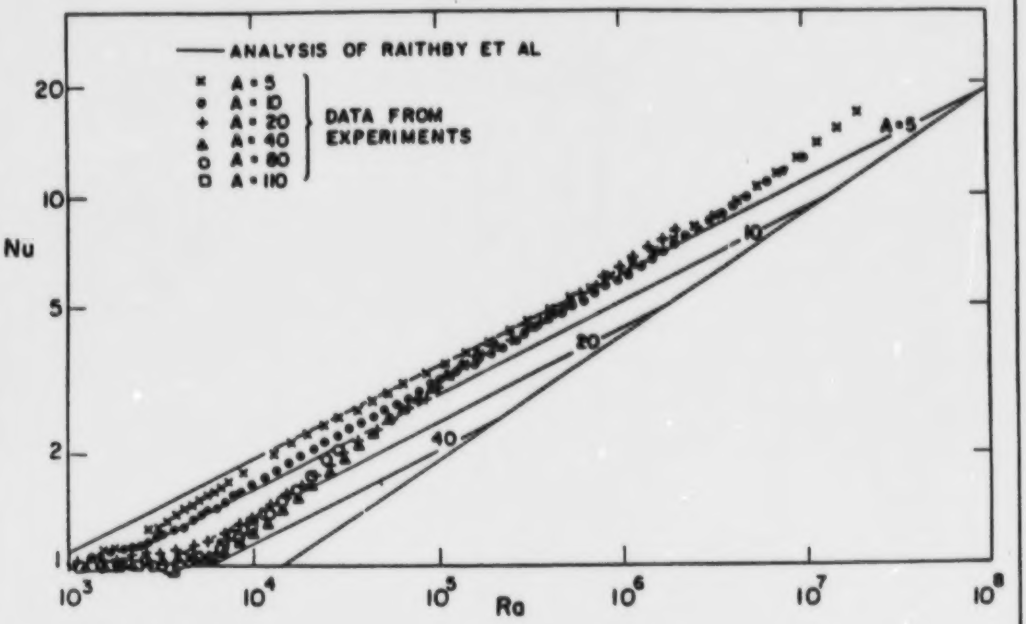


Figure 6.4: Nu vs. Ra and A Data of ElSherbiny, Raithby and Hollands(16) for Air in a Vertical Cavity

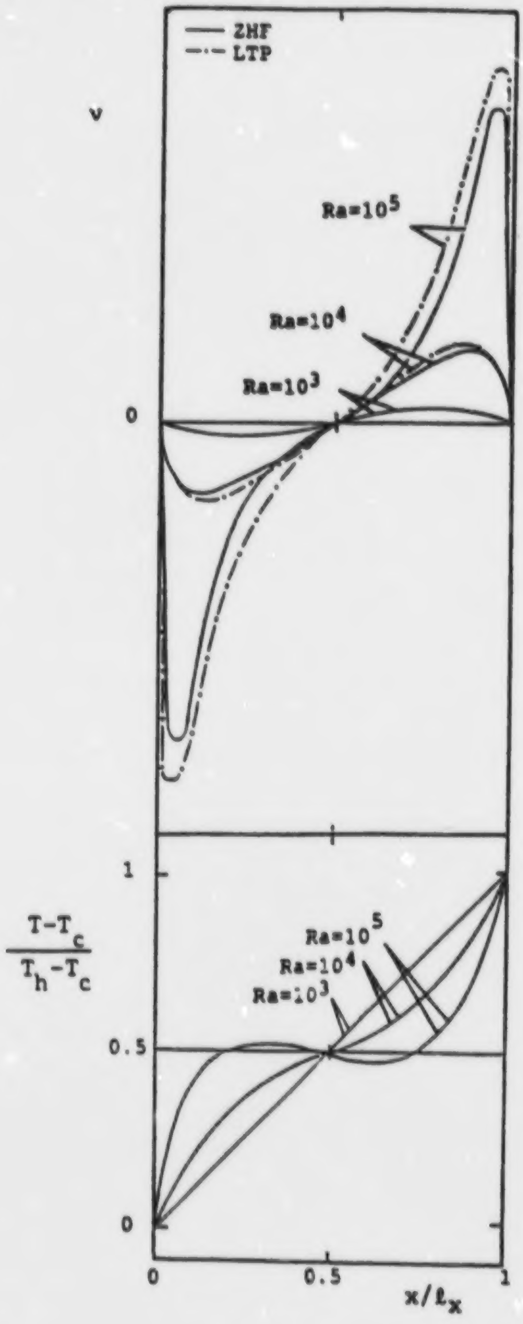


Figure 6.5: Computed Mid-Height Velocity and Temperature Profiles of ElSherbiny, Fath and Refai (78), Vertical Air Layer, A=1

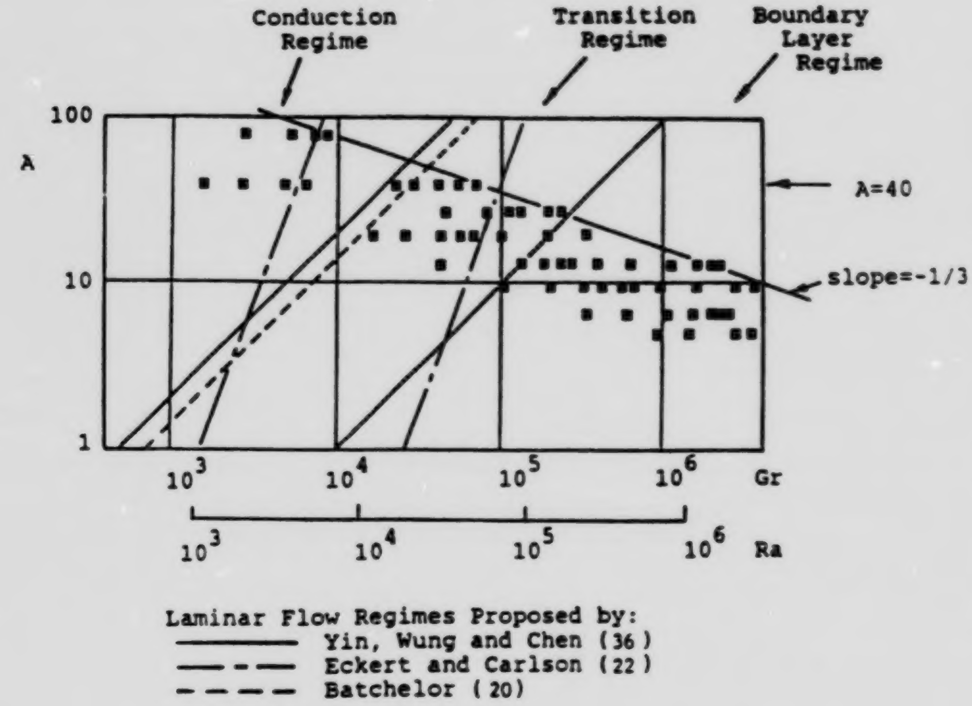


Figure 6.6: Laminar Flow Regimes (from reference (36))

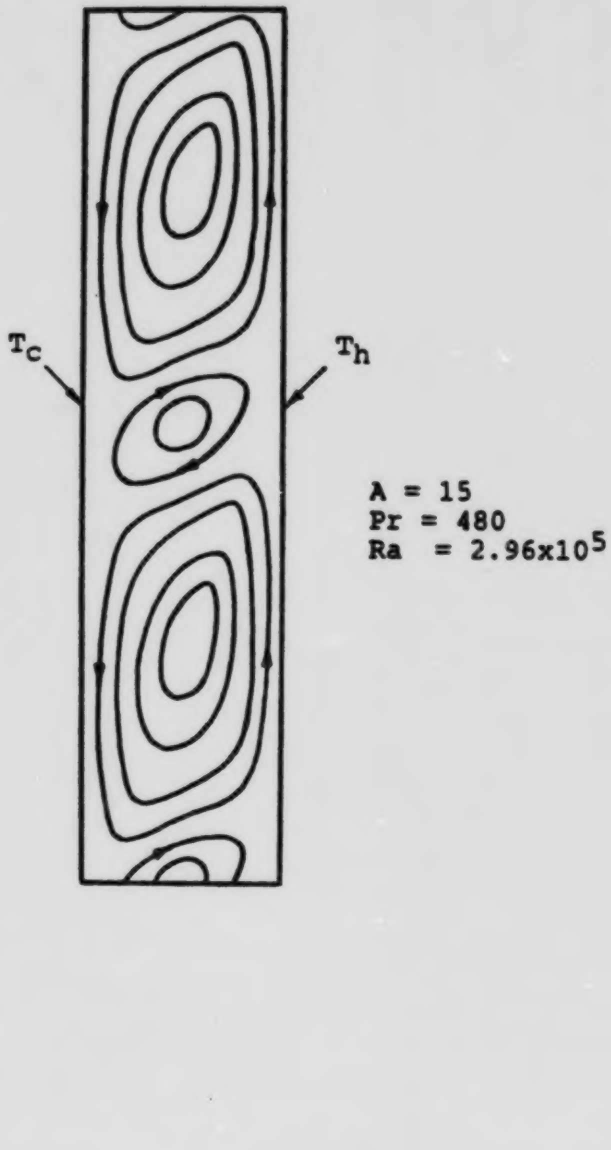
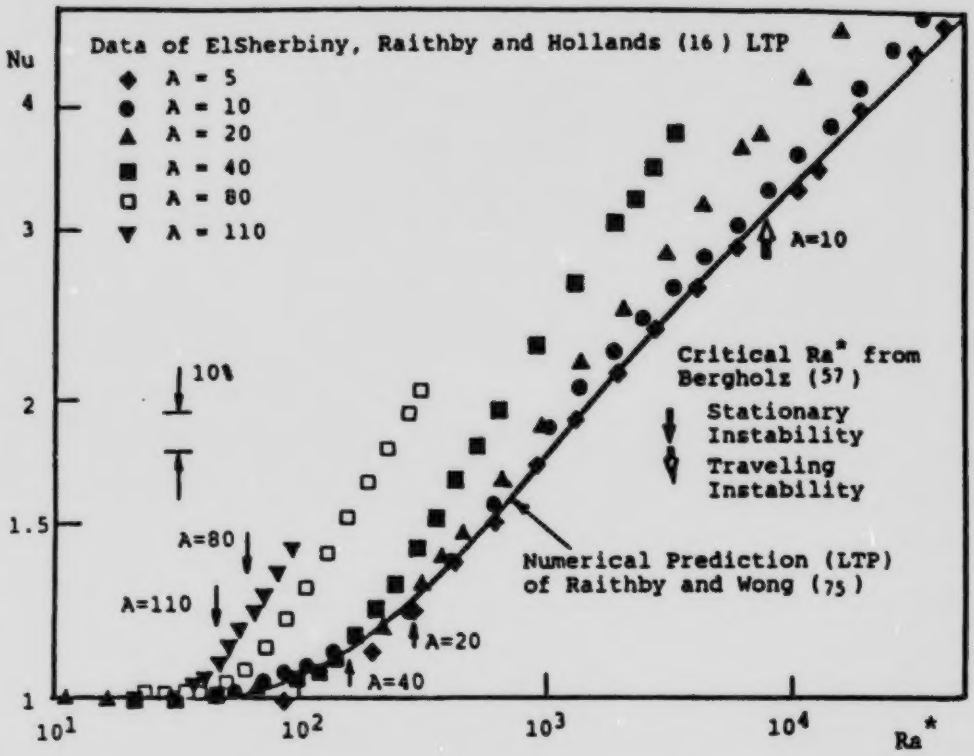

T_c
T_h
A = 15
Pr = 480
Ra = 2.96×10^5

Figure 6.7: Sketch of Secondary and Tertiary Flow (based on reference (59))



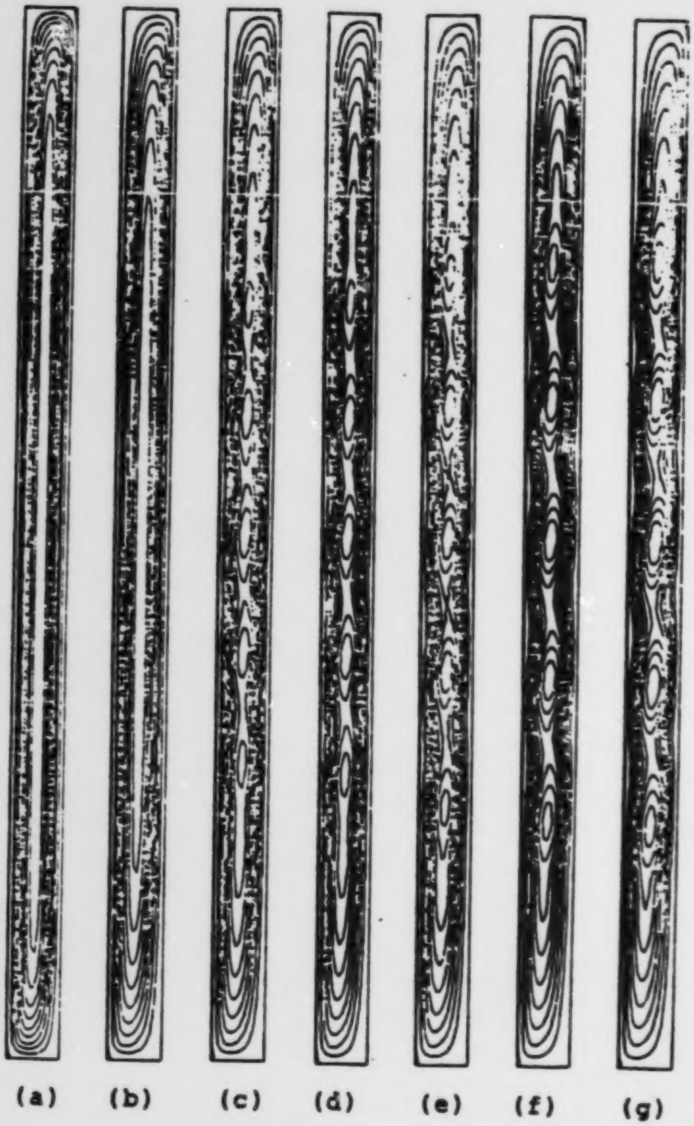
$Ra^* = C \cdot Ra$

$C = (1 - \frac{1.02}{A^{0.44}})/A$

at A=40

$Ra^* = 0.02Ra$

Figure 6.8: Comparison of Predictions of Raithby and Wong (75) (solid curve) with Experimental Data (16)



Ra = (a) 3.55×10^3
(b) 7.10×10^3
(c) 7.81×10^3
(d) 8.52×10^3
(e) 1.07×10^4
(f) 1.42×10^4
(g) 1.78×10^4

Figure 6.9: Numerical Streamline Solutions of Lee and Korpela (76) Showing Secondary Convective Cells in Air, A=20

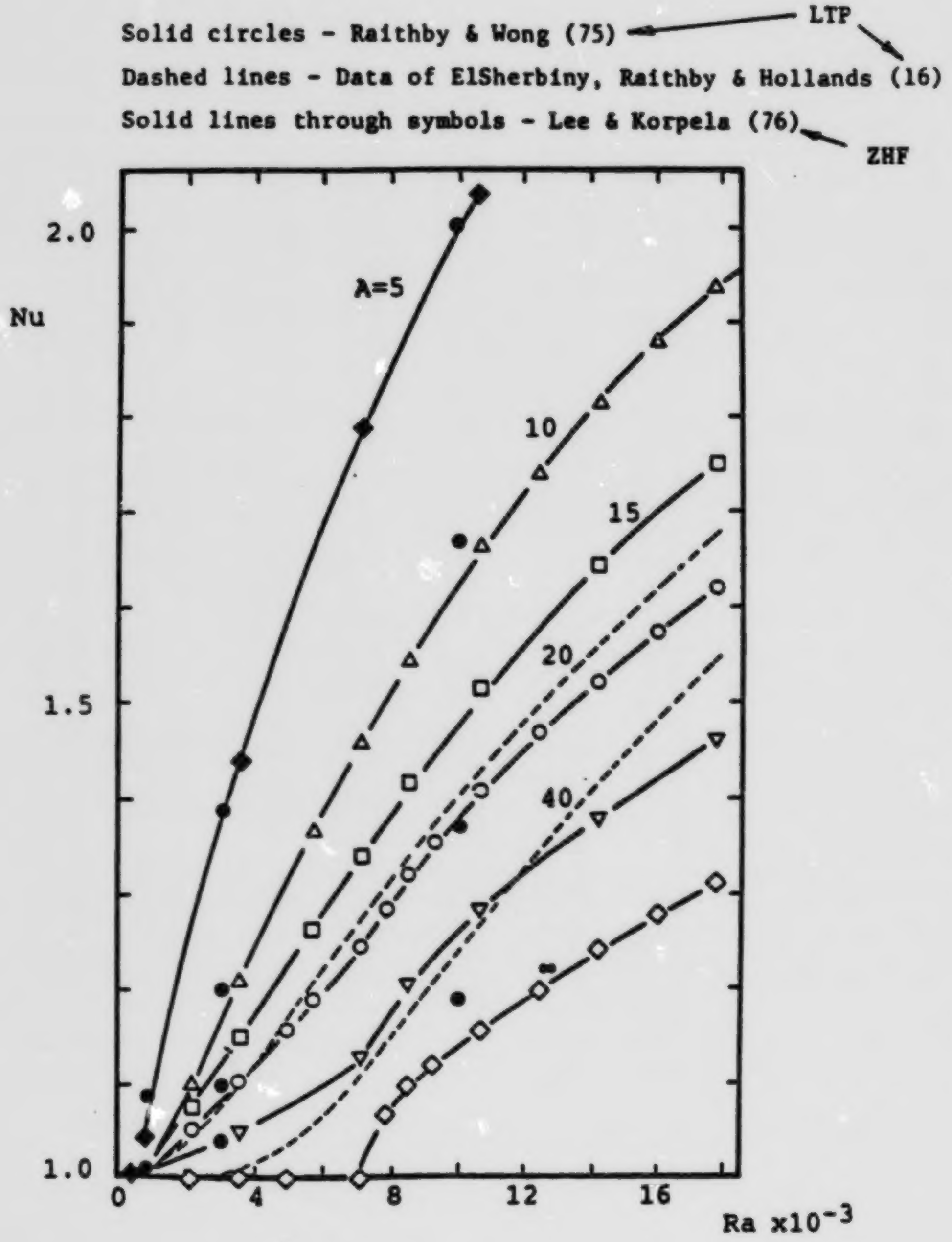


Figure 6.10: Comparison of Predictions of Lee and Korpela (76) with Experimental Data (16)

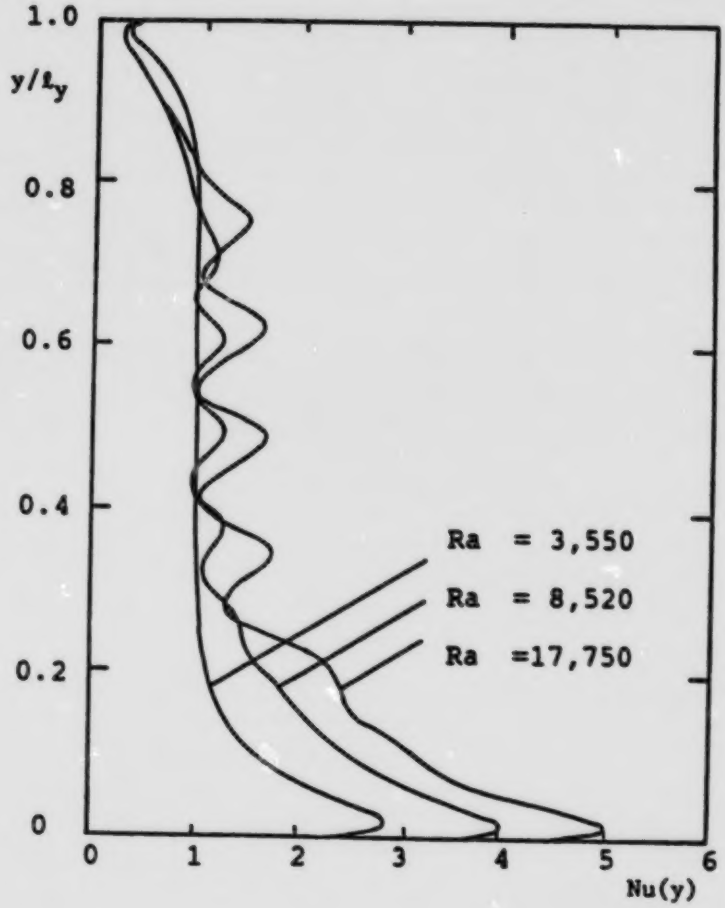


Figure 6.11: Local Nusselt Number, Nu(y) for A=20 Based on Data from Korpela, Lee and Drummond (79)

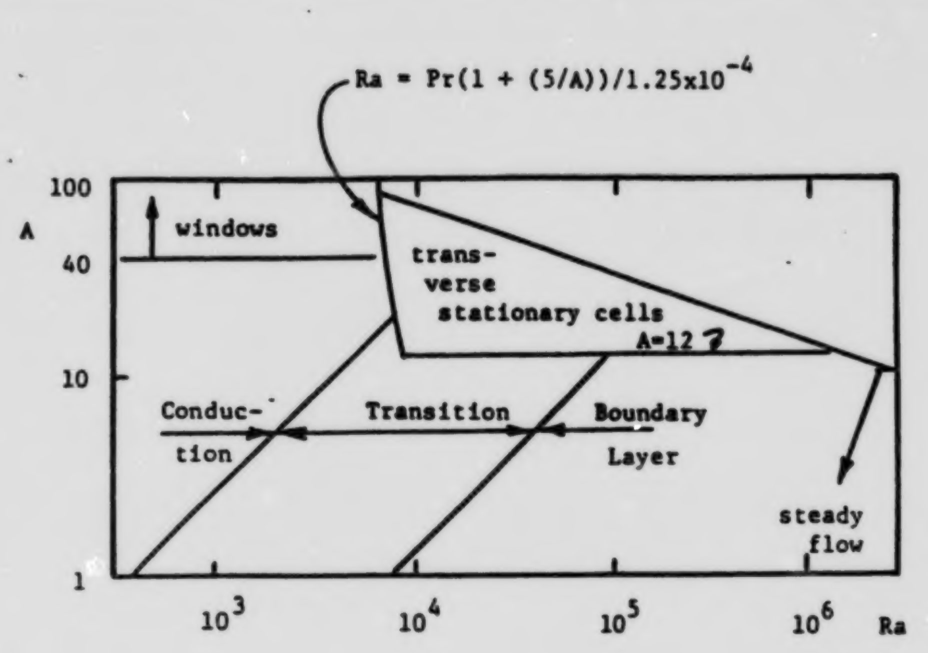


Figure 6.12: Convective Flow Regimes for Vertical Window Cavities

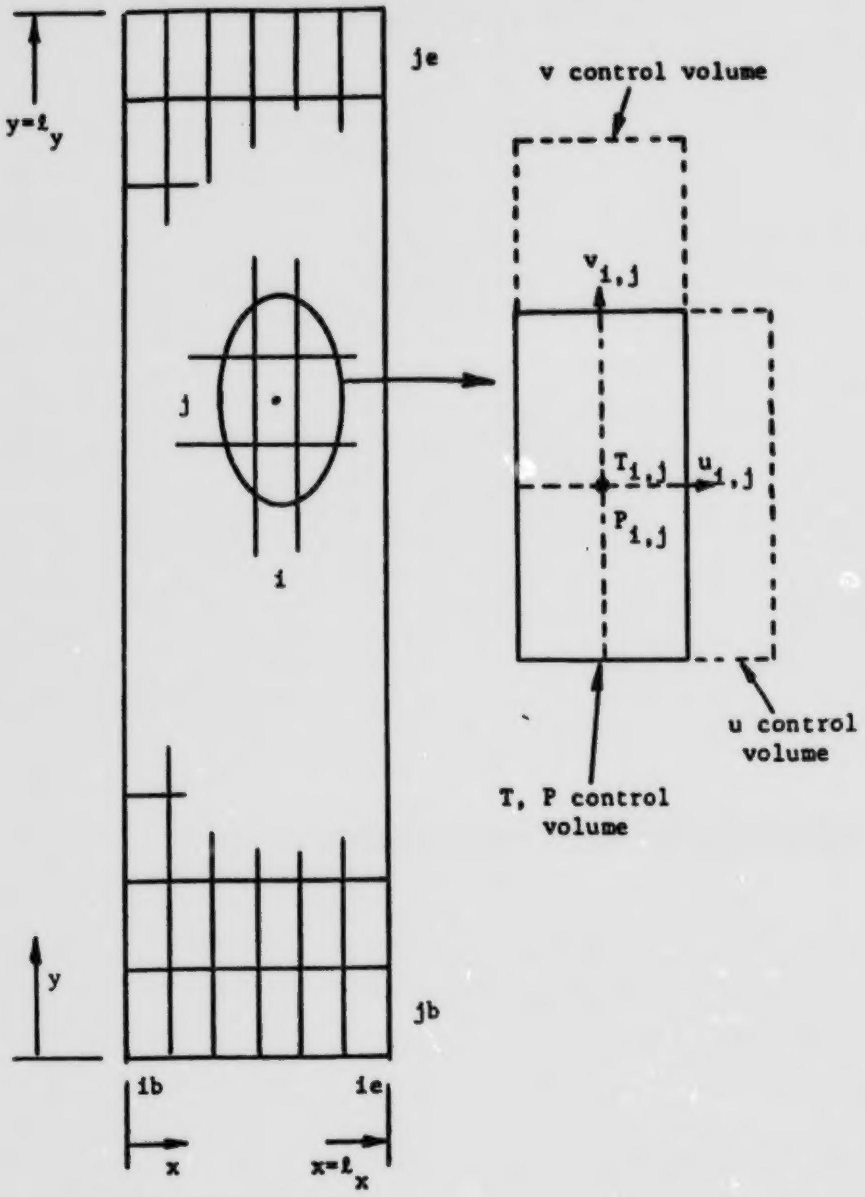

je
y=ℓ_y
v control volume
v_{i,j}
j
T_{i,j}
u_{i,j}
P_{i,j}
i
u control volume
T, P control volume
y
jb
ib
ie
x
x=ℓ_x

Figure 6.13: The Uniform, Staggered Grid

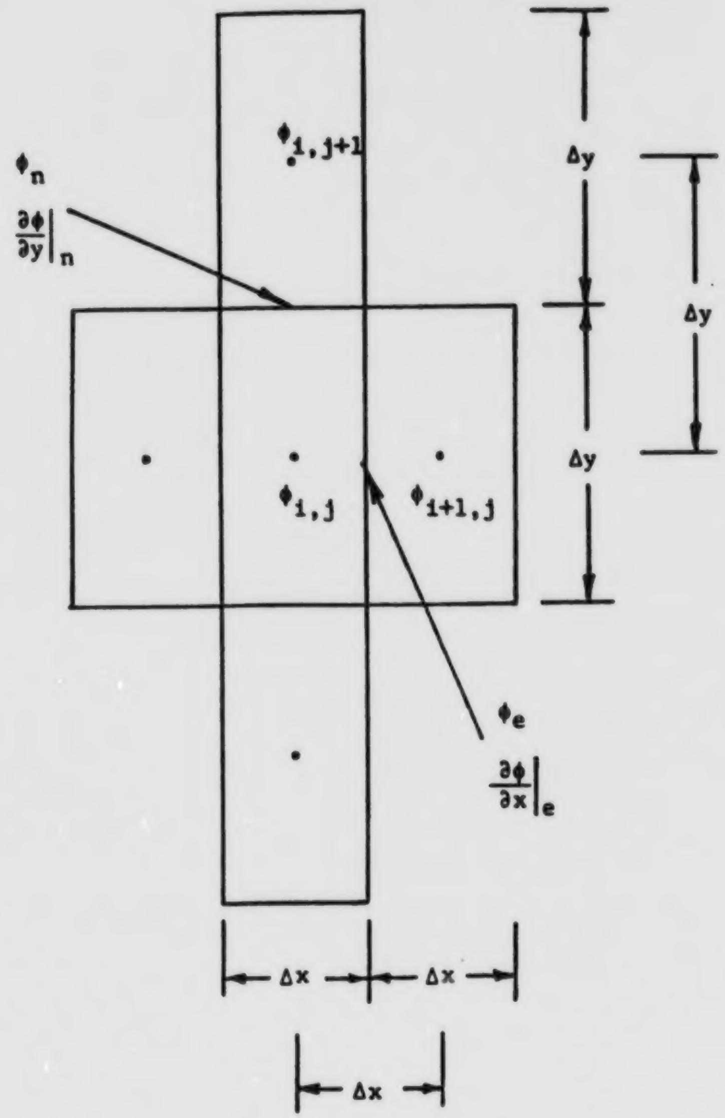

ϕ_n
$\left.\frac{\partial\phi}{\partial y}\right|_n$
$\phi_{i,j+1}$
Δy
Δy
Δy
$\phi_{i,j}$
$\phi_{i+1,j}$
ϕ_e
$\left.\frac{\partial\phi}{\partial x}\right|_e$
Δx
Δx
Δx

Figure 6.14: Calculation of Fluxes at Faces in the Finite Volume Computational Molecule

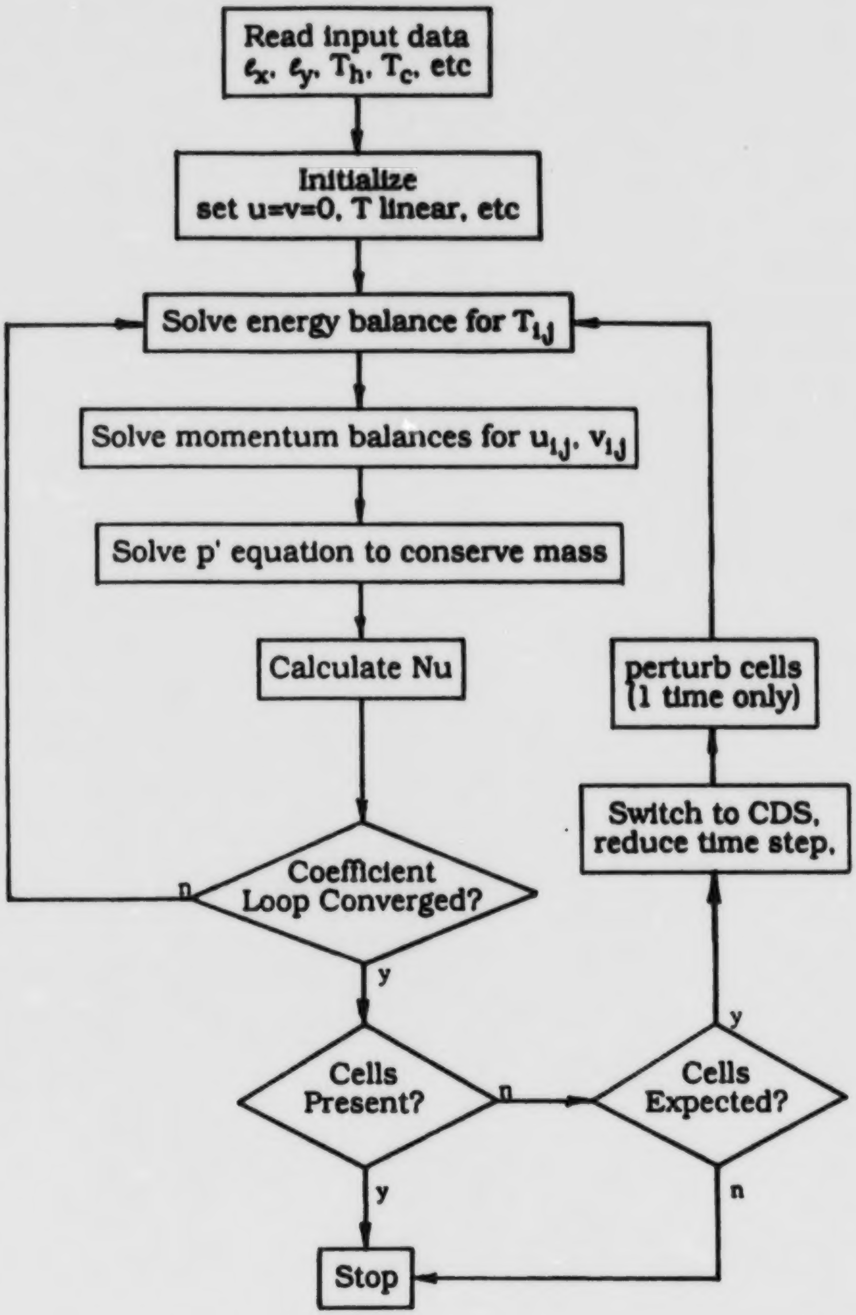

Read input data
ℓ_x, ℓ_y, T_h, T_c, etc
Initialize
set u=v=0, T linear, etc
Solve energy balance for $T_{i,j}$
Solve momentum balances for $u_{i,j}$, $v_{i,j}$
Solve p' equation to conserve mass
Calculate Nu
Coefficient Loop Converged?
n
y
Cells Present?
n
y
Cells Expected?
y
n
Switch to CDS, reduce time step.
perturb cells (1 time only)
Stop

Figure 6.15: Solution Algorithm Flow Chart

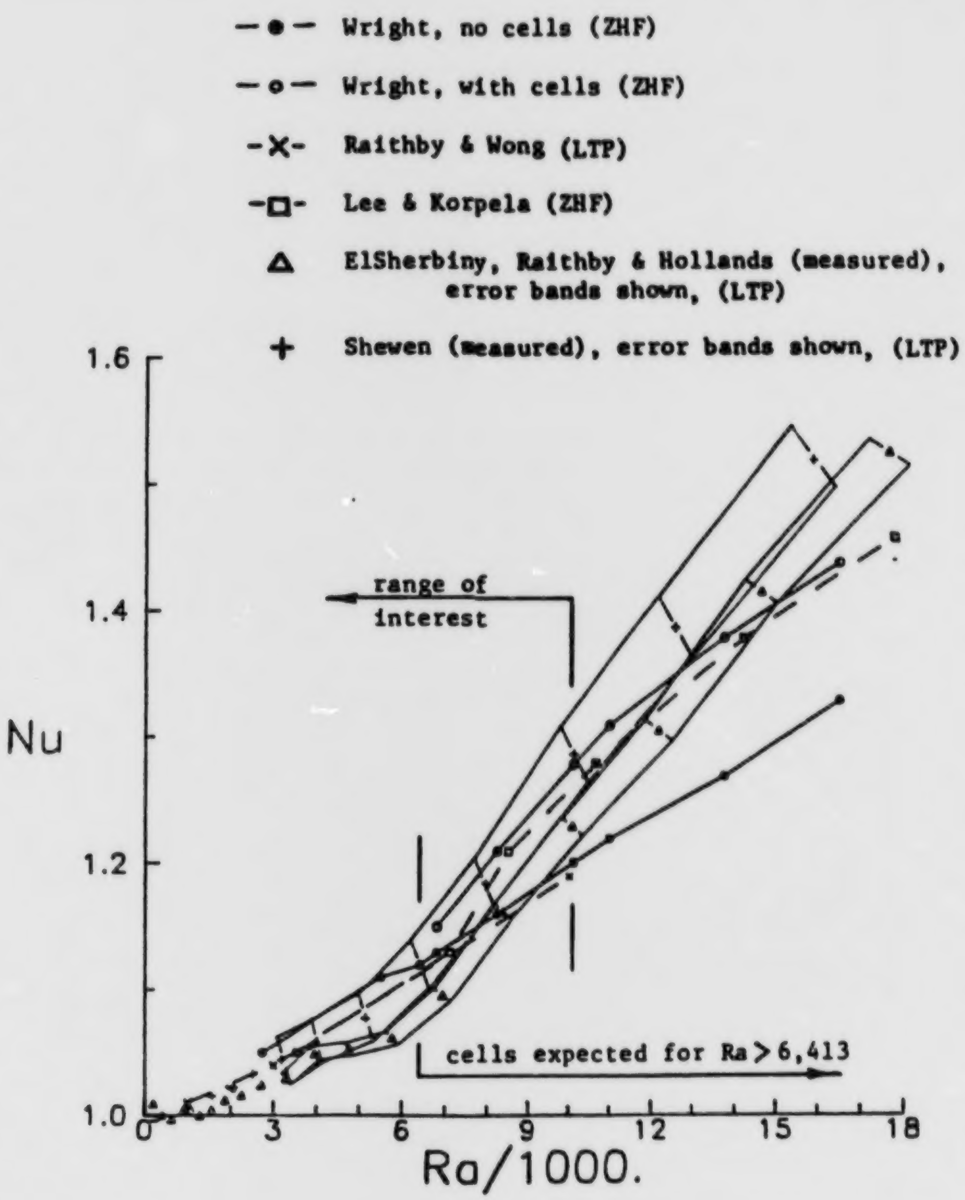
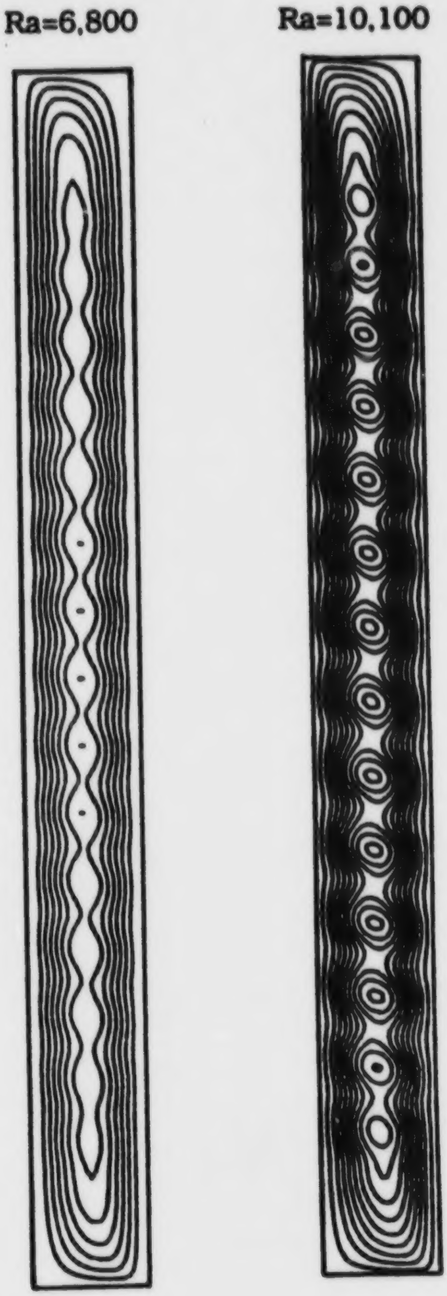


Figure 6.16: Comparison of Available Simulation and Measured Nu vs. Ra Results for the Vertical Cavity, A=40



Note:
Horizontal
dimension
expanded by a
factor of 4

Figure 6.17: Stream Function Plots at Ra=6,800 and Ra=10,100
Line Spacing = 0.02 m^3/sec, A=40

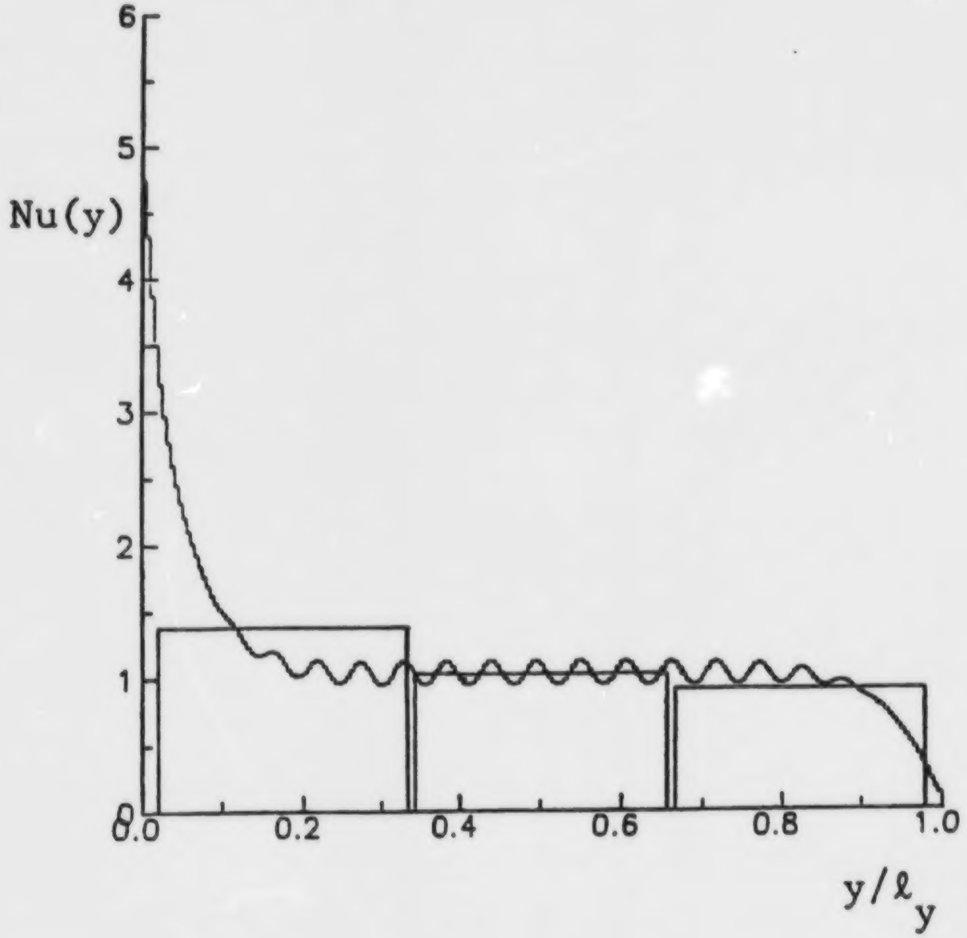


Figure 6.18: Nu(y) versus y and Averaged Over Metering Areas of ElSherbiny, Ra=6,800, A=40

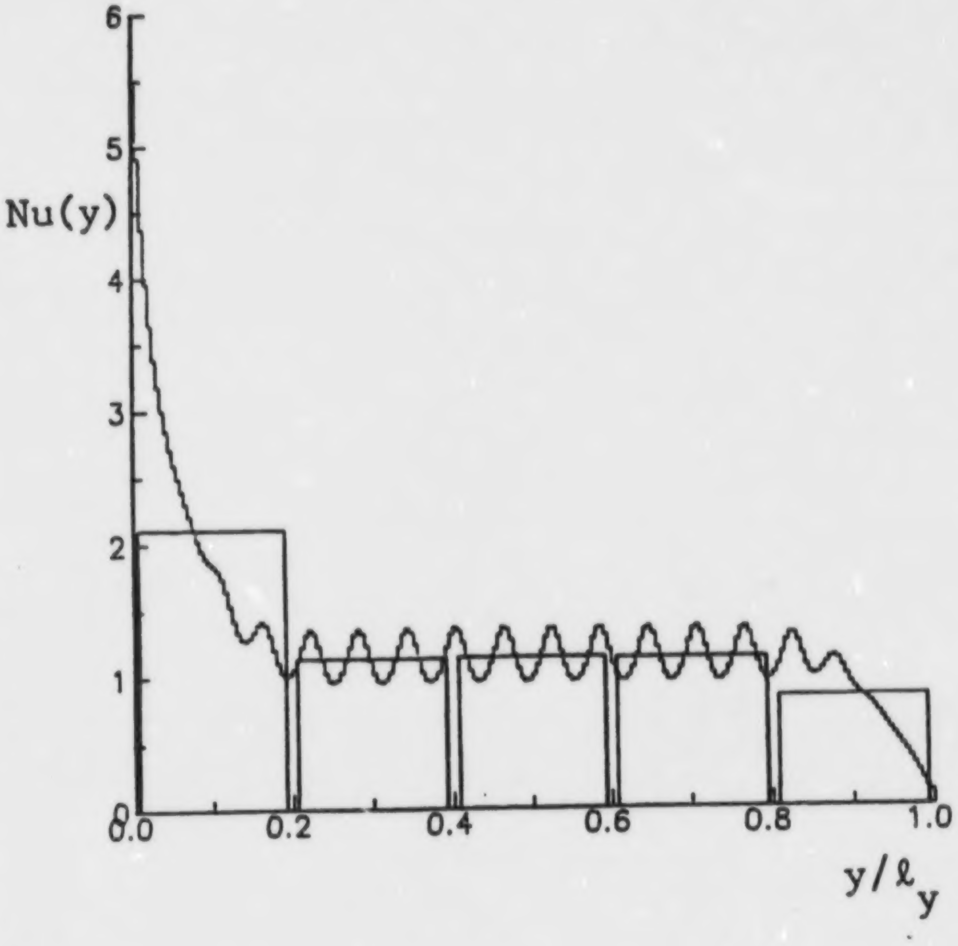


Figure 6.19: Nu(y) versus y and Averaged Over Metering Areas of Shewen, Ra=10,100, A=40

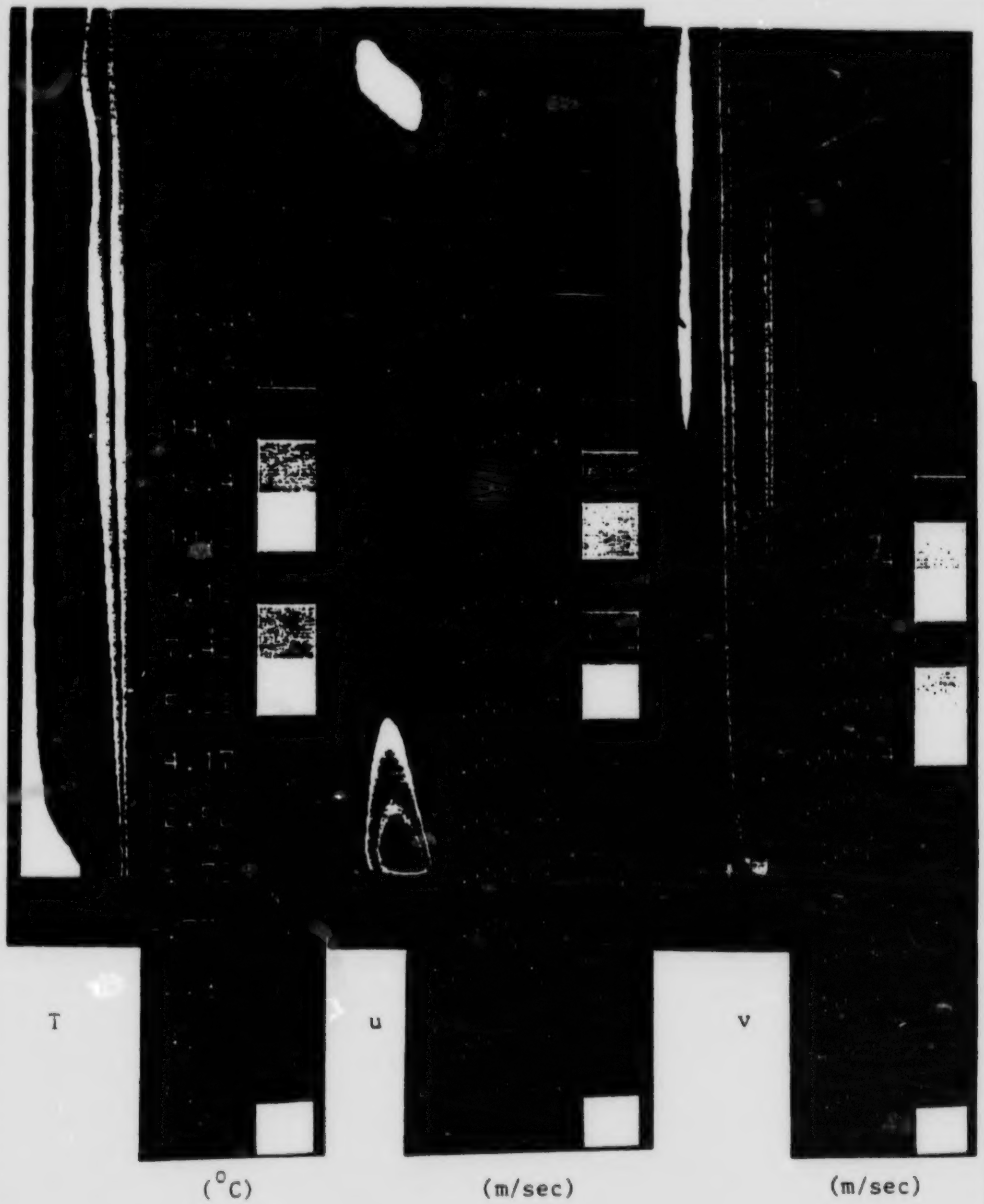


Figure 6.20: PATRAN Fringe Plots of T, u, v at Bottom End of Vertical Slot, Ra=6,800 and A=40

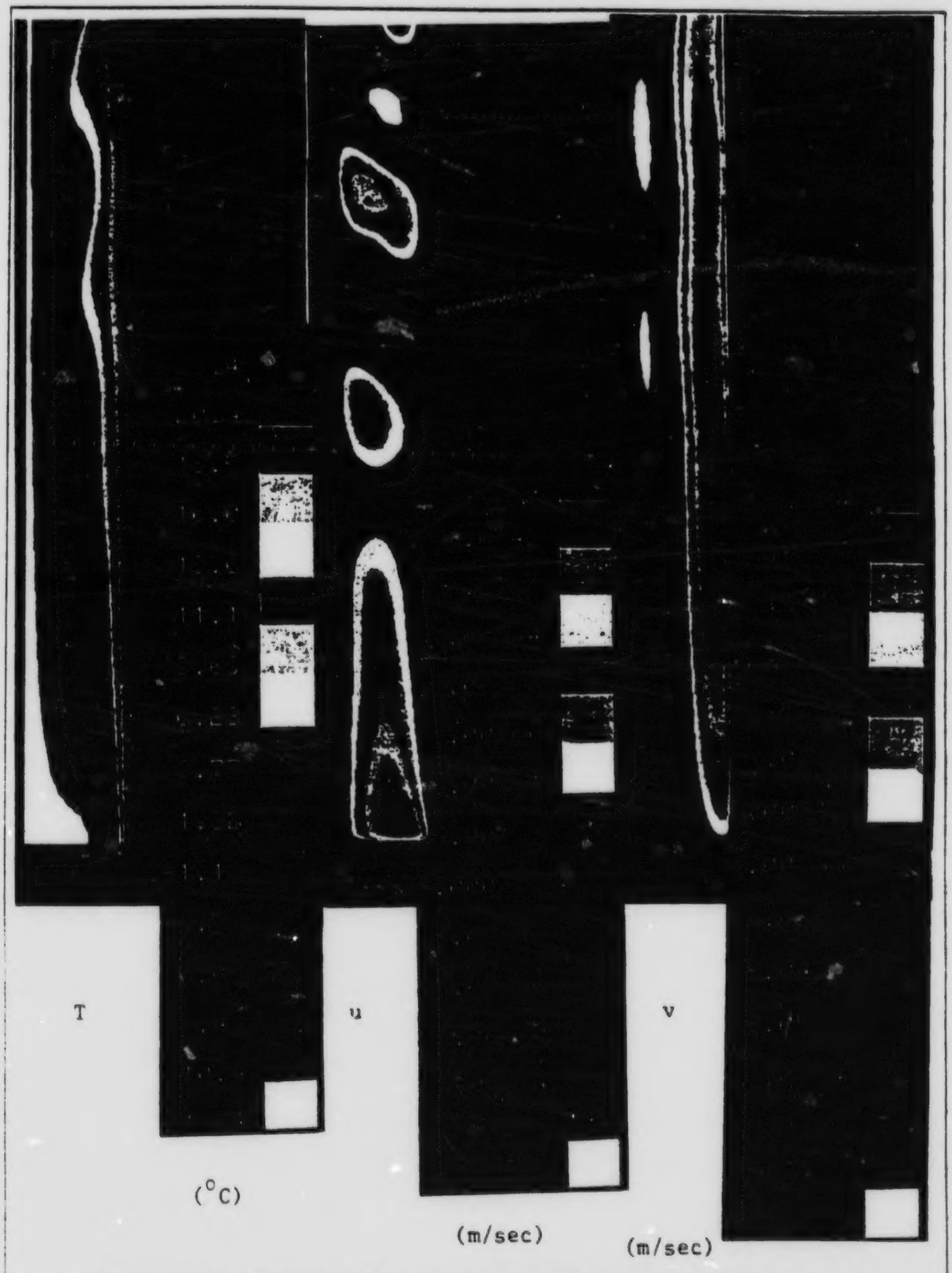


Figure 6.21: PATRAN Fringe Plots of T, u, v at Bottom End of Vertical Slot, Ra=10,100 and A=40

Chapter 7

2-D NUMERICAL MODEL FOR GLAZING SYSTEM THERMAL ANALYSIS

APPROACH

Having successfully developed a 2-D numerical procedure for simulating natural convection in glazing cavities, the model was extended to solve for the temperature and heat flux distributions in double pane glazing system assemblies. Modifications required to model this composite problem domain involved the additional consideration of conductive heat transfer in the solid sections (i.e., glazings and edge-seals) and the exchange of energy by thermal radiation between the surfaces facing the inter-pane cavity.

The 2-D glazing system model was configured to simulate the conditions of a guarded heater plate test using the UWNCA. Details of the problem domain and pertinent nomenclature are shown in Figure 7.1. The glazings are separated from the isothermal boundaries (i.e., the copper plates) by two sheets of neoprene rubber and the top and bottom ends of the problem domain are assumed to be adiabatic. The purpose of performing simulations for the guarded heater plate test condition was to enable a direct comparison between calculated and measured, average and local, heat transfer rates. The analysis can easily be changed to more closely approximate "real world" conditions.

SOLID SECTIONS

The solid sections added to the problem domain consist of two sheets of neoprene, two sheets of glass and two edge seals. Each solid was treated as a homogeneous material.

The control volume arrangement was modified to incorporate the glazing system materials. The x-direction index, i, begins and ends at $i=ib$ and $i=ie$, respectively. The ib and ie control volumes correspond to the layers of neoprene. The glazings are located at $i=ib+1$ and $i=ie-1$.

The cavity includes control volumes from i=ibc=ib+2 to i=iec=ie−2. The edge-seals are uniformly split into the same number of horizontal divisions as the interpane cavity (i.e., 25 columns of control volumes). The grid aspect ratio in the cavity was again set at $A_{grid}=5$ and the same vertical subdivision used in the cavity was used for the glazing and neoprene materials adjacent to the cavity. The edge-seals were divided into n_{seal} vertical divisions such that the size of the seal control volumes was about the same as the cavity control volumes (usually $n_{seal}=5$). The same vertical divisions were used to set up control volumes in the glazings and neoprene adjacent to the edge-seals. The y-direction index, j, runs from j=jb to j=je over the entire problem domain and from je=jbc to j=jec over the height of the cavity.

Adaptation of the solution procedure to model the composite problem was not complicated. The SIMPLEC algorithm was left largely unaltered. The primary modification was in repositioning the boundary conditions. The energy balance was applied over the entire problem domain with the thermal boundary conditions (See Figure 7.1) applied at the outer edges. This step in the coefficient update loop generated a temperature solution for all control volumes. However, the velocity boundary conditions, used in solving balances to conserve momentum and mass, were applied at the walls of the cavity. Thus, the natural convection velocity field in the cavity was generated for non-uniform temperature distributions at the cavity walls and all velocities at locations outside the cavity were set to zero.

Application of the energy balance across the composite problem domain required care to ensure that the heat flux was correctly calculated at surfaces between materials of different thermal conductivities. Fortunately, such surfaces only exist where the velocity is zero and additional book-keeping was only required to keep track of heat transfer by conduction. The conductive heat flux at such control volume faces was calculated according to the procedure outlined by Patankar (1). For example, at a surface on the west wall of the interpane cavity, as shown in Figure 7.2, the surface temperature, T_{surf}, is given by:

$$T_{surf} = \{T_{ibc-1,j}(k_g/t_g) + T_{ibc,j}(k/\Delta x)\} \,/\, \{(k_g/t_g) + (k/\Delta x)\} \qquad (7.1)$$

Note that at any surface, the upwinding coefficients α and β are equal to 0 and 1, respectively, because the velocity at the wall is zero.

The corresponding surface heat flux, q_{surf}, is given by:

$$q_{surf} = 2(k_g/t_g)(T_{ibc-1,j} - T_{surf}) = 2(k/\Delta x)(T_{surf} - T_{ibc,j}) \qquad (7.2)$$

THERMAL RADIATION MODEL

Figure 7.1 shows the fill gas cavity. Each of the four cavity walls can be divided into finite surfaces - each of which emits, reflects and absorbs thermal radiation. The incident radiative flux (or irradiance) at the p^th^ surface is denoted H_p and can be expressed as:

$$H_p = \sum_{vnb} A_{nb}E_{nb\text{-}p}B_{nb}/A_p \qquad (7.3)$$

where the summation is carried out for all of the "visible neighbour" wall surfaces (i.e., for all surfaces from which radiation can reach surface p). A_{nb} is the surface area of a neighbour surface, $E_{nb\text{-}p}$ is the geometric exchange factor from a neighbour surface to surface p and B_{nb} is the radiative flux leaving (or radiosity at) the neighbour surface. Using the reciprocity relation:

$$A_{nb}E_{nb\text{-}p} = A_pE_{p\text{-}nb} \qquad (7.4)$$

Equation 7.3 can be simplified.

$$H_p = \sum_{vnb} E_{p\text{-}nb}B_{nb} \qquad (7.5)$$

If all surfaces are treated as diffusely emitting and reflecting then the exchange factors, $E_{p\text{-}nb}$, are equal to the more familiar geometric radiative shape factors, $F_{p\text{-}nb}$. However, it was thought possible that an appreciable portion of the radiation reflected from a glass surface would be reflected in a specular manner - especially if a low-e coating were present. It has been shown that subdivision of the hemispheric reflectance into specular and diffuse components, coupled with the assumption of diffuse emission, lends itself to analytical treatment (2). In accordance with this method the reflectances of the vertical walls were split into specular and diffuse components as shown in Equation 7.6.

$$\rho = 1 - \varepsilon = \rho^s + \rho^d \tag{7.6}$$

Using this formulation for reflectance, B_{nb}, can be expressed as the sum of emitted flux (surface temperature = $T_{surf,nb}$) and the diffusely reflected portion of the irradiance.

$$B_{nb} = \varepsilon_{nb}\,\sigma\,(T_{surf,nb})^4 + (\rho_{nb})^d\,H_{nb} \tag{7.7}$$

Combining Equations 7.5 and 7.7 and rearanging:

$$H_p - \sum_{vnb} E_{p\text{-}nb}\,(\rho_{nb})^d\,H_{nb} = \sum_{vnb} E_{p\text{-}nb}\,\varepsilon_{nb}\,\sigma(T_{surf,nb})^4 \tag{7.8}$$

Equation 7.8 can be expressed, for all surfaces of the cavity, in matrix format.

$$[A][H] = [T] \tag{7.9}$$

where the transpose of [H] is given by

$$[H]^t = (H_1, H_2, H_3 \;\; \ldots\ldots\ldots\ldots \;\;) \tag{7.10}$$

and the entries of the pth row of the coefficient matrix, [A], are given by the left hand side of Equation 7.8. The pth entry of the column vector, [T], is given by the right hand side of Equation 7.8. The solution to Equation 7.10 can be expressed in terms of the inverse of [A].

$$[H] = [A]^{-1}[T] \tag{7.11}$$

Notice that [A] is a function of optical properties and geometry but does not depend on surface temperatures. It needs to be solved only once for a given cavity and selected set of surfaces. (In fact, the computer code does not invert the [A] matrix, but reaches the same end by calculating its LU decomposition. Details can be found in Chapter 2 of reference (3).) The [T] matrix is a function of the surface temperatures. Therefore, for a given distribution of surface temperatures [T] can be generated and the solution for [H] using Equation 7.11 requires little computer time.

The specular reflectance of the edge spacers was assumed to be negligible. In windows this surface is customarily oxidized aluminum or a tar-like material. The surface of more innovative edge spacers might be fiberglass or foam.

The exchange factors, $E_{p\text{-}nb}$, also need only be calculated once for a given geometry. Significant simplification in the calculation of $E_{p\text{-}nb}$ follows from the fact that ρ^s is constant over all of the individual surface subdivisions that comprise each of the walls and from the assumption that ρ^s can be set to zero for the horizontal walls. Consider the calculation required if surface p is on the west wall of the cavity and the neighbour surface being considered is on the opposite wall. Figure 7.3 shows this geometry. It can be seen that $E_{p\text{-}nb}$ is:

$$E_{p\text{-}nb} = F_{p\text{-}nb} + \sum_{i=1}^{\infty} (\rho_W{}^s)^i \, (\rho_E{}^s)^i \, F(2i)_{p\text{-}nb} \qquad (7.12)$$

where $\rho_W{}^s$ and $\rho_E{}^s$ are the specular reflectance values for the west and east vertical walls. $F(2i)_{p\text{-}nb}$ is the geometric shape factor from surface p to the mirror image of surface nb after 2i reflections in the vertical walls. Each $F_{p\text{-}nb}$ was computed using Hottel's crossed string rule (4) and Equation 7.12 (or one of several similar equations) was applied by including the first 100 terms in the summation. Knowing that all radiation leaving any surface must eventually be absorbed or diffusely reflected at some surface, the accuracy of the calculated exchange factors was verified using the following check-sum.

$$\sum_{\forall nb} E_{p\text{-}nb}(1-\rho_{nb}{}^s) = 1 \qquad \text{for any surface, p} \qquad (7.13)$$

If the fill gas cavity is bounded by n grey surfaces on each vertical wall and m surfaces on each horizontal wall, and their temperatures are taken as being fixed, then the application of Equation 7.8 at each surface requires the solution of 2(n+m) simultaneous equations - yielding 2(n+m) values of H_p. This procedure also entails the calculation and storage of $2(n+m)^2$ values of $E_{p\text{-}nb}$. It was recognized that the computing requirements would be unnecessarily excessive if the surfaces specified for the radiation solution directly matched the control volume grid used for the convection and conduction portions of the analysis. (i.e., $n \approx 250$, $m \approx 25$).

Fortunately, there is a large section away from the ends of the cavity (the centre-glass section) where the temperatures of the vertical walls were expected to be isothermal. It was assumed that the centre-glass portion of each glazing could be treated as a single surface for the purpose of applying Equation 7.8. These large centre-glass surfaces were delimited according to the ASHRAE definition of centre-glass area; exposed faces of finite control volumes starting at more than 65 mm from the sight line were included in the centre-glass surface. In the edge-glass portions of the cavity the individual exposed faces of the control volumes were used in the analysis of radiation exchange. This approach reduced the number of H_p variables by a factor of about 4 and reduced the required storage for $E_{p\text{-}nb}$ by a factor of more than 10.

Once the irradiance at each enclosure surface was known the net flux of radiant energy into the surface, S_p, could be found.

$$S_p = \varepsilon_p(H_p - \sigma(T_{surf,p})^4) \tag{7.14}$$

S_p was included as a fixed energy source in the energy balance of the wall control volumes by way of the source term, S, in Equation 6.2. The rate of energy generation per unit volume at any wall control volumes is:

$$S = S_p/\Delta n \tag{7.15}$$

where Δn is the dimension of the control volume normal to the wall. The validity of the source terms was examined during each simulation by summing energy generation at the cavity wall control volumes. This total should be zero.

S_p and S were updated at each iteration of the coefficient loop after the generation of a new temperature field.

SOLUTION CONVERGENCE

The 2-D glazing system simulation software was coded in a Fortran program called IGU-2D. The program listing is included in Appendix D. Aside from allowing for repositioned boundary conditions and thermal radiation source terms at the cavity walls the solution algorithm incorporated in IGU-2D differs very little from the algorithm used for the analysis of the vertical slot natural convection problem. However, the criterion for solution convergence in IGU-2D is based on the calculation of total heat transfer through the glazing system, Q, rather than the Nusselt number of the cavity.

The SGU was subdivided into 29 columns of control volumes. Therefore, it was possible to calculate Q at 30 vertical cuts in the problem domain. Once the converged solution has been reached these estimates of Q should be equal because adiabatic surfaces were prescribed at the top and bottom boundaries. Q was calculated at the right hand edge of the i^{th} column of control volumes using:

$$Q = Q_{rad} + \sum_{j=jb}^{je} q_{i,j}\Delta y_j \qquad (7.16)$$

where, Q_{rad} is the net amount of thermal radiation energy crossing the vertical cut from right to left and can be determined by summing the rate of energy generation due to source terms at wall control volumes to the left of the vertical cut.

and, $q_{i,j}$ is the non-radiative, right-to-left, heat flux through the right hand face of the $(i,j)^{th}$ control volume. This heat flux, for vertical control volume faces within the cavity, is given by:

$$q_{i,j} = \beta k(T_{i+1,j} - T_{i,j})/\Delta x - \rho C_p u_{i,j}\{(0.5+\alpha)T_{i,j} + (0.5-\alpha)T_{i+1,j}\} \qquad (7.17)$$

where, α and β are the upwinding coefficients given in Chapter 6.

In the case of control volume faces not within the cavity $q_{i,j}$ was calculated according to Equation 7.2.

During each execution of IGU-2D the rate of heat transfer was calculated at the 30 vertical cuts and the maximum and minimum of these values, Q_{max} and Q_{min}, respectively, were noted. Iteration of the coefficient update loop was terminated when Q_{max} and Q_{min} differed by less than 0.1%.

PRELIMINARY RESULTS

Four preliminary simulation runs were carried out in order to examine the sensitivity of the solution to the specular/diffuse split chosen for the long wave reflectivities of the vertical cavity surfaces. Two glazing systems were modeled: a conventional double glazed unit and a similar unit with a soft low-e coating. Both glazing units had corrugated strip edge-seals and the fill gas was air. Two simulations were run for each glazing system. Highly specular reflectance properties were used in one simulation and purely diffuse reflectances were used in the other. In the specular runs the glass surfaces were modeled with $\rho^s=0.15$ and $\rho^d=0.01$; the low-e surfaces with $\rho^s=0.90$ and $\rho^d=0.01$. In the diffuse runs the glass surfaces were modeled with $\rho^s=0.0$ and $\rho^d=0.16$; the low-e surfaces with $\rho^s=0.0$ and $\rho^d=0.91$.

The output from each pair of simulations showed very little or no difference in all reported quantities. For example, the local heat fluxes calculated for the areas corresponding to the locations of the guarded heater plates differed at most in the fourth significant digit. Plotted heat flux and temperature profiles appeared to be identical. It was concluded that the diffuse/specular long wave reflection split at the glazing cavity surfaces has no significant bearing on heat transfer in glazing units. All surfaces were subsequently modeled as purely diffuse reflectors.

COMPARISON WITH MEASURED HEAT FLUX

Seven glazing systems were tested using the guarded heater plate apparatus and modeled using IGU-2D. All except one had low-e coatings. Two units had argon fill gas. The set of edge-seals incorporated in the test units was purposely chosen to span a wide range of thermal resistance. They included the highly conductive conventional (aluminum spacer bar) edge-seals, corrugated strip edge-seals with intermediate thermal resistance plus spacers made from strips of rigid insulating foam. The various data needed to describe the glazing systems are listed in Table 7.1.

In order to simulate glazing systems installed in the UWNCA the top and bottom edges of each glazing system were treated as adiabatic surfaces. When measurements were undertaken these same surfaces were well insulated and it was assumed that any heat flux through the ends of the glazing system would be very small in relation to the heat flux between the glazings through the edge-seals. Thus, measurement errors due to the non-adiabatic end condition were assumed to be negligible. However, this may not have been the case for measurements taken when highly insulating foam spacers were used. The same foam that was used for edge-spacers was also used to insulate the edge of the test section. Therefore, even though the possibility of a calculation/ measurement mismatch was foreseen it was decided that the foam spacers should be simulated because of (a) the insight to be gained by examining the heat flux patterns in systems with highly insulated edge-seals and (b) the chance that the non-adiabatic experimental boundary condition was unimportant or that its impact might be quantified.

Figure 7.4 shows a sample solution (glazing unit # 3) of heat flux at the right hand edge of the problem domain (i.e., at the vertical surface where $T=T_h$). This surface corresponds to the face of the hot copper plate in the guarded heater plate apparatus. The heat flux from the hot copper plate is shown as a function of distance from the bottom of the glazing unit. Glazing unit #3 was built with conventional edge-seals that contain aluminum spacer bars. Figure 7.4 shows that the low thermal resistance of this edge-seal results in a very high heat flux (in excess of 600 W/m^2) near the top and bottom edges of the glazing unit. Heat flux through the centre-glass area is constant except for the small variation caused by the secondary cells. The asymmetry between the top and bottom edge-glass areas is a result of the fill gas motion.

Table 7.1:
Calculated and Measured Glazing System Heat Flux Results

	ΔT_{pp}	e_x	w_{seal}	k_{seal}	t_n	t_g	ε_{le}	ε_{seal}	q_{meas}	q_{calc}
	[°C]	[mm]	[mm]	[W/ m·K]	[mm]	[mm]			[W/m^2] Top/Middle/ Bottom	
1)	18.88	12.13	9.53	0.52	3.175	3.94	–	0.827	96.9	90.1
									95.3	89.4
	fill gas air, no coatings								105.7	96.7
2)	19.39	12.41	9.53	0.53	3.175	2.84	0.096	0.827	49.0	45.1
									47.0	44.1
	fill gas air								58.9	55.0
3)	19.12	14.12	11.05	2.56	1.524	3.3	0.047	0.054	32.1	33.5
									28.7	28.0
	fill gas argon								45.7	45.6
4)	11.38	11.96	11.3	2.12	3.175	3.91	0.094	0.054	24.4	24.5
									19.7	19.9
	fill gas argon								27.67	27.1
5)	27.42	16.0	25.4	0.035	1.524	5.59	0.07	0.579	56.5	44.9
									62.7	56.1
	fill gas air								89.0	87.2
6)	27.34	13.0	25.4	0.035	1.524	5.59	0.07	0.579	60.8	50.2
									61.2	59.6
	fill gas air								83.7	82.6
7)	14.86	13.0	25.4	0.035	1.524	5.59	0.07	0.579	32.3	26.6
									30.6	30.2
	fill gas air								38.7	38.7

Notes: 1) Calibration of neoprene mats: k_n=0.17 W/m·K for 1.524 mm mats and k_n=0.19 W/m·K for 3.175 mm mats.

2) k_{seal} was based on seal dimensions and k_{lin}=0.41 W/m·K (corrugated strip seals) for units 1 and 2; k_{lin}=2.0 W/m·K (conventional seals) for units 3 and 4. Units 5, 6 and 7 had rigid insulating foam spacers without sealant and k_{seal}=0.035 W/m·K was taken directly from the manufacturer's literature.

3) Emissivities of seals, ε_{seal}, and low-e coatings, ε_{le}, were measured using a Gier-Dunkel DB-100 Infrared Reflectometer.

Figure 7.4 also shows three average values of the calculated heat flux corresponding to the locations of the guarded heater plates where heat transfer was experimentally metered. These results can be directly compared to measured heat flux results. The medium/low/high progression of heat flux found at the top/middle/bottom heater plates is typical of measured values taken for many glazing units tested using the UWNCA. The middle heater plate area provides a representative measure of centre-glass heat flux. The heat flux from the top and bottom heater plates is augmented by the thermal short-circuit at the edge-seal and the motion of the fill-gas further increases the heat flux from the bottom heater plate.

Two sets of heater plate area heat flux results are included in Table 7.1 for each glazing system – one set calculated and one set measured. Each set consists of heat flux data for the top, middle and bottom heater plates. The same results are presented in bar-chart format in Figure 7.5.

Units 1 and 2

The first two glazing units for which results are shown in Figure 7.5 were built using corrugated strip edge-seals. The fill gas was air. Unit 1 had no coatings but unit 2 had a soft low-e coating. Secondary cells were not expected and were not perturbed for either of these glazing units. These units were tested at about the same temperature difference but unit 2 allowed about half as much heat transfer as unit 1 because of the low-e coating. The difference between heat flux at the centre and bottom heater plates was greater for unit 2 because the low-e coating suppresses centre-glass heat transfer but does not alter the thermal resistance of the edge-seals.

The calculated heat flux results consistently underpredicted the measured data from units 1 and 2 by about 7%. This level of discrepancy is not unreasonable but the fact that it exists for both units and is independent of heater plate location suggests that improvement might be possible if the reason(s) for the difference were known. In addition to showing good agreement with the measured heat flux data the trend in the calculated results very closely follows the top/middle/bottom heater plate trend that was expected and exhibited in the measurements.

Units 3 and 4

Glazing units 3 and 4 were both built with conventional, aluminum spacer bar, edge-seals and both contained argon fill-gas. Unit 4 had a conventional soft low-e coating. Unit 3 had the more recently developed soft low-e coating with extremely low emissivity. Secondary cells were modeled in unit 3 but not in unit 4 which was tested at a lower temperature difference.

The agreement between calculated and measured heat flux in these glazing units was excellent. The heater plate heat flux was predicted to within 2% in 5 of the 6 cases.

Arrows marked in Figure 7.5 indicate the difference in calculated heat flux resulting from the presence of secondary cells. The results for unit 3 show that Ra was low enough that the secondary cells had little impact on the heat flux. The presence of cells augmented the heat flux at the top and middle heater plates by about 4% but caused little change (a very slight decrease) at the bottom heater plate.

Again, the customary top/middle/bottom trend in heat flux was observed because of the highly conductive edge-seals. Note that the bottom heater plate heat flux was about 50% higher than the heat flux at the middle heater plate for units 3 and 4. The corresponding difference for units 1 and 2 was only about 25% because they were built with the less conductive corrugated strip edge-seals.

Unit 3 was tested with about the same temperature difference that was used to test units 1 and 2. It can be seen, by examining the heat flux at the centre heater plate that the thermal resistance at the centre of the test section was approximately doubled by the addition of the low-e coating. The subsequent use of argon fill gas almost doubled the thermal resistance again.

Units 5, 6 and 7

Test units 5, 6 and 7 each contained a soft low-e coating and had air fill-gas. These three units did not incorporate commercial edge-seals. Instead, the glazings were separated at their edges by strips of rigid, insulating foam. No sealant was used. Thus, the edge-seal consisted of a single material with an extremely low thermal conductivity.

Units 5, 6 and 7 do not exhibit the top/middle/bottom trend in heat flux usually found in glazing systems because the low conductivity spacers do not drive the heat flux up drastically at the top and bottom. Consequently, the heat flux at the top heater plate is about the same as, or may even be less than, the heat flux at the middle heater plate. It is interesting that the creation of cells in units 5 and 6 decreased the heat flux at the bottom heat plate.

The calculated and measured heat fluxes for unit 5 do not agree as well as the results for any of the other units. The heat flux measured at the centre heater plate was 23% higher than the heat flux calculated without secondary cells. This discrepancy was cut in half after cells were introduced. The reason for the disparity can be understood by noting that the relatively large pane spacing and high temperature difference resulted in Ra being well in excess of the range of Ra for which the model was intended ($Ra \approx 16{,}000$ based on centre-glass temperature difference across the cavity). On the basis of results shown in Figure 6.1 it can be seen that the numerical model would underpredict Nu by almost 20% at $Ra = 16{,}000$ if no cells were perturbed and that only half of the difference would be made up if cells were perturbed.

The simulation results for units 6 and 7 very closely match the measured heat fluxes for the bottom and middle heater plates. In these locations discrepancies were 3% or less. In contrast, the heat flux at the top heater plate was underpredicted by 21% in both units 6 and 7. Notice that the top heat flux was also underpredicted by about 25% for unit 5. This difficulty in calculating heat flux at the top heater plate exists only for units 5, 6 and 7 – the units with highly insulating edge-spacers. The specific reason for this is not clear. It is likely that this problem is linked to the adiabatic boundary assumption. Fortunately, the model does not exhibit the same difficulty when more conventional edge-seals are used or at the bottom of the glazing system where the heat flux and temperature profiles are of greatest interest.

HEAT FLUX IN THE GLAZING CAVITY

Having completed the comparison of calculated and measured heat fluxes at the problem boundary it is instructive to study the simulation results for details about heat flux within the glazing unit itself. Test units 1, 2, 3 and 7 were chosen for examination because they include glazing systems with and without a low-e coating, with and without argon fill gas plus all three variations in edge-seal conductance. Units 1, 2 and 3 were tested at approximately the same temperature difference facilitating the comparison of heat flux results. The simulation results for units 1, 2, 3 and 7 were assumed to be accurate on the basis of the comparisons presented in the previous section.

Figures 7.6, 7.7, 7.8 and 7.9 show the local x-direction heat flux, q(y), plotted as a function of distance from the bottom of the glazing system for a vertical surface located at the hot wall of the cavity (i.e., $x=t_n+t_g+\ell_x$). Figures 7.6 through 7.9 show results for test units 1, 2, 3 and 7, respectively. Each figure includes two curves; the higher curve showing total heat flux and the lower curve showing the radiative component. The hash marks on the lower curve delimit the centre-glass portion of the glazing system as specified in the analysis of radiative exchange which in all cases corresponds to the centre-glass area defined by ASHRAE. The sight lines are located where the radiative heat flux becomes zero and the total heat flux rises sharply.

Figure 7.6 shows heat flux results for a conventional double glazed SGU with corrugated strip edge-seals (unit 1). The heat flux is constant over the centre-glass area and radiative exchange constitutes the majority of the centre-glass heat transfer. In the bottom edge-glass area the heat flux increases because of the fill gas motion. Heat flux through the edge-seals peaks at more than 400 W/m^2 and is significantly higher than the heat flux through any portion of the sight area.

Unit 2 was similar to unit 1 except that a low-e coating was included in unit 2. The results for these two units can be compared by examining Figures 7.6 and 7.7. The pattern of heat flux is similar for units 1 and 2 but the total centre-glass and edge-glass heat flux has been substantially reduced because the low-e coating has virtually eliminated the radiative heat flux in unit 2. In contrast, the peak edge-seal heat flux is greater in unit 2 (in excess of 500 W/m^2) than it was in unit 1. It is interesting that the increased edge-seal heat flux results from the presence of the low-e coating. The increase in edge-glass and centre-glass thermal resistance imposes a larger temperature difference across the edge-seals.

Units 2 and 3 can be compared by examining the results in Figures 7.7 and 7.8. The differences between these units include a switch from corrugated strip to conventional edge-seals and the presence of argon fill gas instead of air. Again the low-e coating limits the radiative heat flux to a small portion of the total. The secondary cells in unit 3 create a ripple in the heat flux distribution. The use of argon in unit 3 has reduced centre-glass and edge-glass heat flux. The use of conventional edge-seals and the relatively high thermal resistance of the sight area have caused the peak heat flux in the edge-seals to be in excess of 1,000 W/m^2.

Figure 7.9 shows heat flux results for unit 7. Unit 7 was similar to unit 2 except for the use of insulating foam instead of the corrugated strip edge-seals. Unit 7 was tested at a lower temperature difference (ΔT_{pp}=14.9°C versus 19.4°C). The centre-glass thermal resistance of unit 7 was approximately the same as it was for unit 2. The effect of the insulating foam spacer used in unit 7 is apparent in that the peak heat flux did not exceed 90 W/m^2 which is an order of magnitude less than the peak heat flux found in unit 3. Away from the cavity wall the edge-seal heat flux drops to almost the level of the centre-glass heat flux. The asymmetry of the heat flux distribution caused by the fill gas motion is highly pronounced in the results shown for unit 7.

Several observations can be made regarding results that are common to Figures 7.6 through 7.9. In each case the radiant heat flux distribution did not show a discontinuity at the junction of the edge-glass and centre-glass areas. This result supports the validity of lumping the centre-glass control volume surfaces together for the purpose of the radiative exchange model. The simulations also give credence to the ASHRAE definition of centre-glass area in that heat flux results were very close to being constant within the centre-glass limits. In all cases, the heat flux profiles were asymmetric with the natural convection of the fill gas causing the bottom edge-glass heat flux to be significantly greater than the centre-glass or top edge-glass heat flux. This asymmetry is most evident in units with highly insulating edge-seals. This augmentation of heat flux at the bottom edge of the sight area corresponds to the location of the minimum indoor glazing temperature where the onset of condensation occurs during cold weather.

TEMPERATURE PROFILES AT THE WARM GLAZING

Temperature profiles, T(y) versus y, have been produced for a plane at the hot vertical wall of the glazing cavity. This surface is the same vertical cut at which heat flux profiles where given in Figures 7.6 to 7.9. Figures 7.10 through 7.13 show temperature profiles for units 1, 2, 3 and 7 which correspond to Figures 7.6 through 7.9, respectively.

Figures 7.10, 7.11 and 7.12 show that the centre-glass temperature drew progressively closer to the hot plate temperature as the centre-glass thermal resistance of the glazing system increased. The same effect is apparent in windows exposed to the environment. Glazing systems with higher thermal resistance exhibit higher indoor glazing temperatures in cold weather and are less susceptible to condensation.

The location of the minimum indoor pane temperature is at (or very near) the bottom sight line. The extreme edges of the indoor glazing may exist at a lower temperature but the glass outside of the sight line is generally not exposed to the indoor air after being installed in a sash. Condensation forms first at the bottom sight line. The sight line temperature predictions resulting from simulation of the guarded heater plate configuration offer no quantitative information regarding windows exposed to more realisitic conditions, but qualitative observations can be made.

Examining Figures 7.9 through 7.13 the minimum temperature at the bottom sight line was consistently lower than the temperature at the top sight line. This difference would not have existed if the motion of the fill gas had not been modeled. The difference between the top and bottom sight line temperatures was less than 1°C in all four cases. This difference would be greater if the glazing system were exposed to the indoor environment rather than being held close to an isothermal plate.

In the three units constructed with commercial edge-seals (see Figures 7.10, 7.11 and 7.12) the temperature of the warm glazing continued to drop along the edge-seal below the bottom sight line. This is consistent with the heat flux profiles shown in Figures 7.6, 7.7 and 7.8 where the heat flux also was also found to decrease across the face of the edge-seal. Only in the glazing system with insulating foam spacers, shown in Figure 7.13, was an increase in temperature found below the bottom sight line.

DISCUSSION

A model has been formulated for the numerical simulation of heat transfer in glazing systems. This model includes the 2-D treatment of glazing and edge-seal conduction, the exchange of thermal radiation between the glazings and natural convection of the fill gas including secondary cells if needed. Simulation results pertain to both the centre-glass and edge-glass areas of the glazing system.

Guarded heater plate measurements have been found to agree well with calculated heat flux values for a number of glazing systems with a wide variety of glazing system and edge-seal components. Simulation results have provided insights regarding the heat flux patterns and temperature profiles inside these glazing systems. The validity of the ASHRAE edge-glass definition has been confirmed. The importance of fill gas motion in determining the minimum indoor pane temperature has been demonstrated.

REFERENCES

(1) Patankar, S.V., "Numerical Heat Transfer and Fluid Flow", Hemisphere Publishing Corp, 1980.

(2) Sparrow, E.M., Cess, R.D., "Radiation Heat Transfer", Augmented Edition, McGraw-Hill.

(3) Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., "Numerical Recipes The Art of Scientific Computing," Cambridge University Press

(4) Siegel, R., Howell, J.R., "Thermal Radiation Heat Transfer", McGraw-Hill Inc..

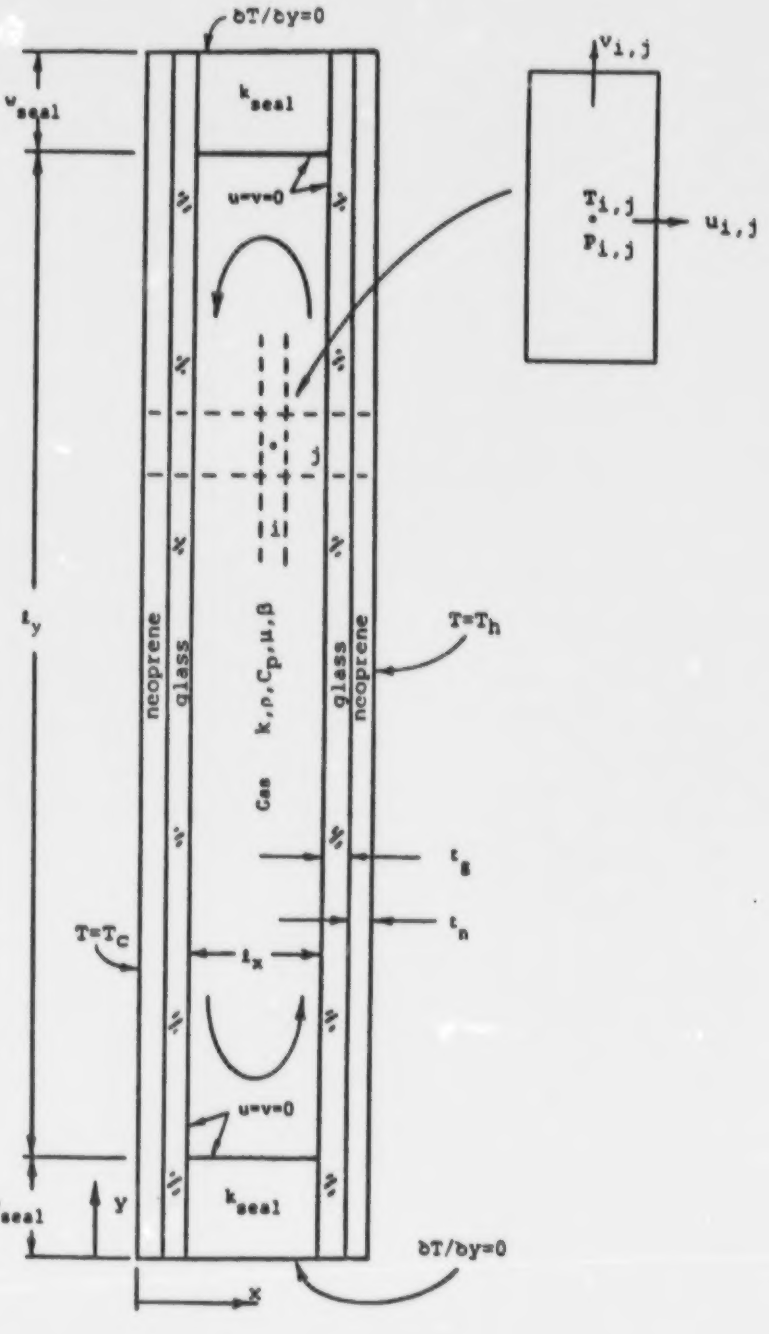

∂T/∂y=0
w_seal
k_seal
u=v=0
v_i,j
T_i,j
P_i,j
u_i,j
j
i
ℓ_y
neoprene
glass
k,ρ,C_p,μ,β
Gas
glass
neoprene
T=T_h
t_g
t_n
T=T_c
ℓ_x
u=v=0
w_seal
y
k_seal
∂T/∂y=0
x

Figure 7.1: 2-D Glazing System Thermal Analysis Geometry

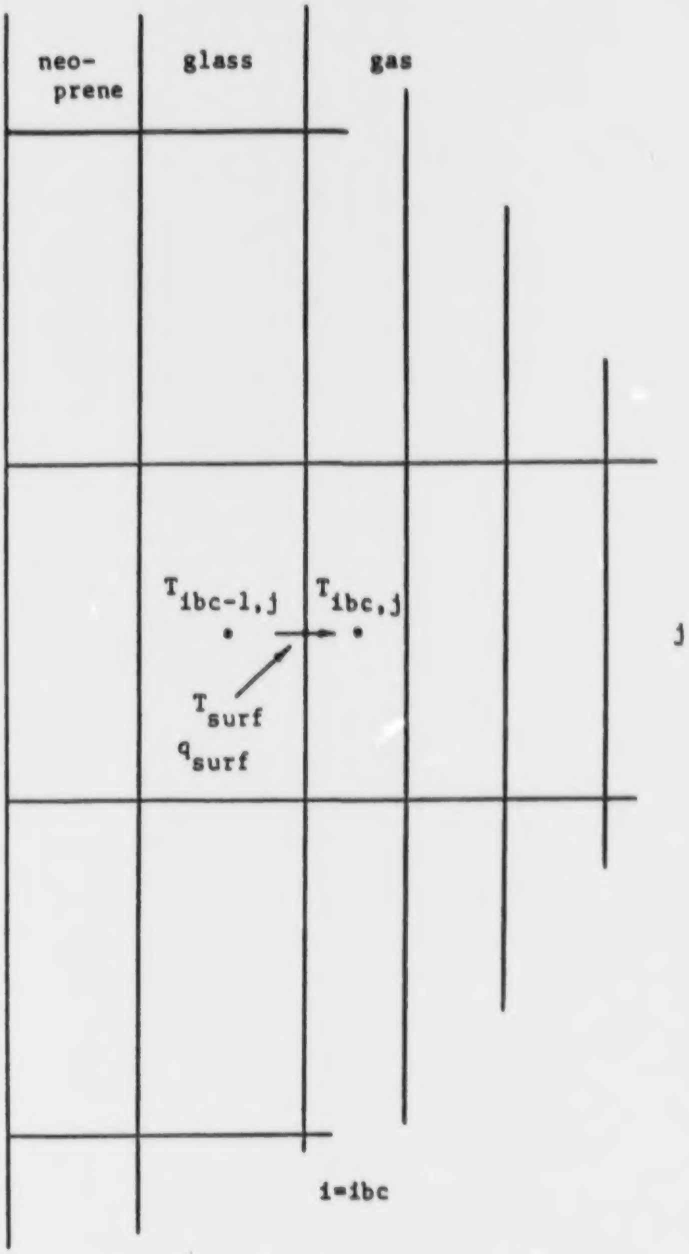

neo-
prene
glass
gas
$T_{ibc-1,j}$
$T_{ibc,j}$
j
T_{surf}
q_{surf}
i=ibc

Figure 7.2: Surface Temperature and Heat Flux at the Cavity Wall

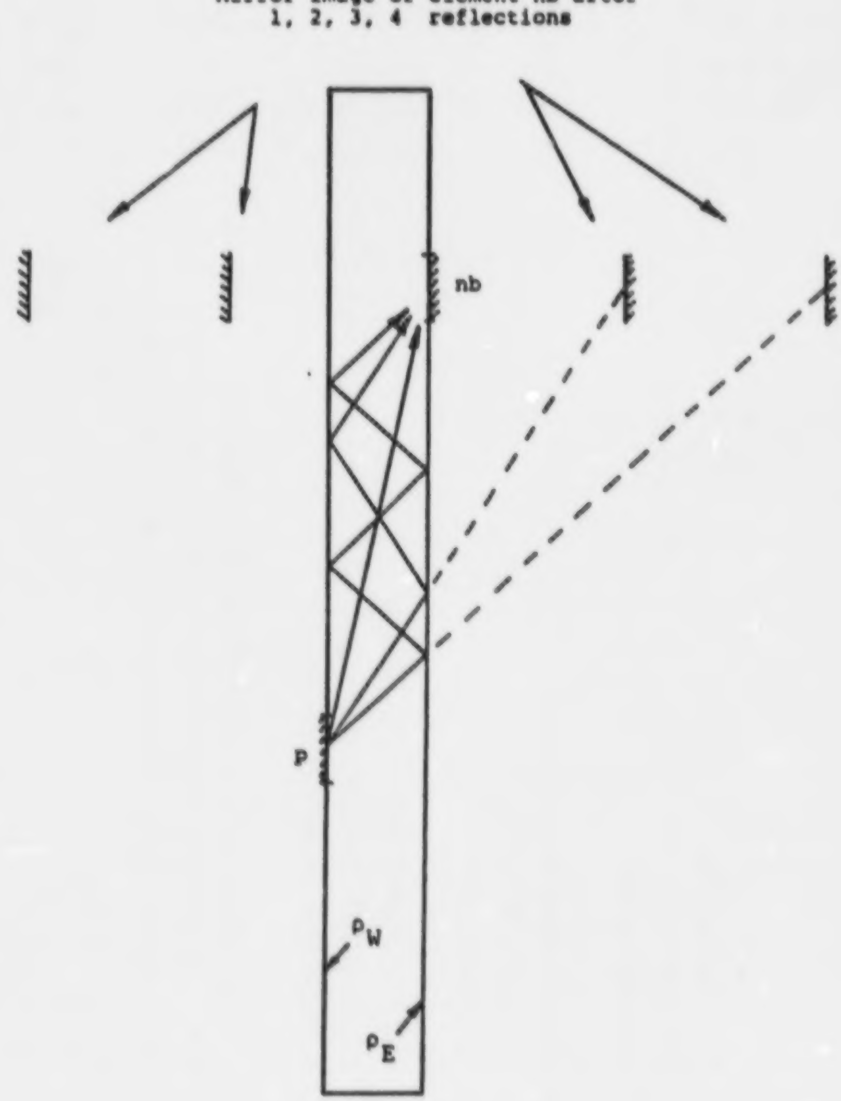

Mirror image of element nb after
1, 2, 3, 4 reflections
nb
P
ρ_W
ρ_E

Figure 7.3: Example of Geometry for Calculation of $E_{p\text{-}nb}$ Radiative Exchange Factor

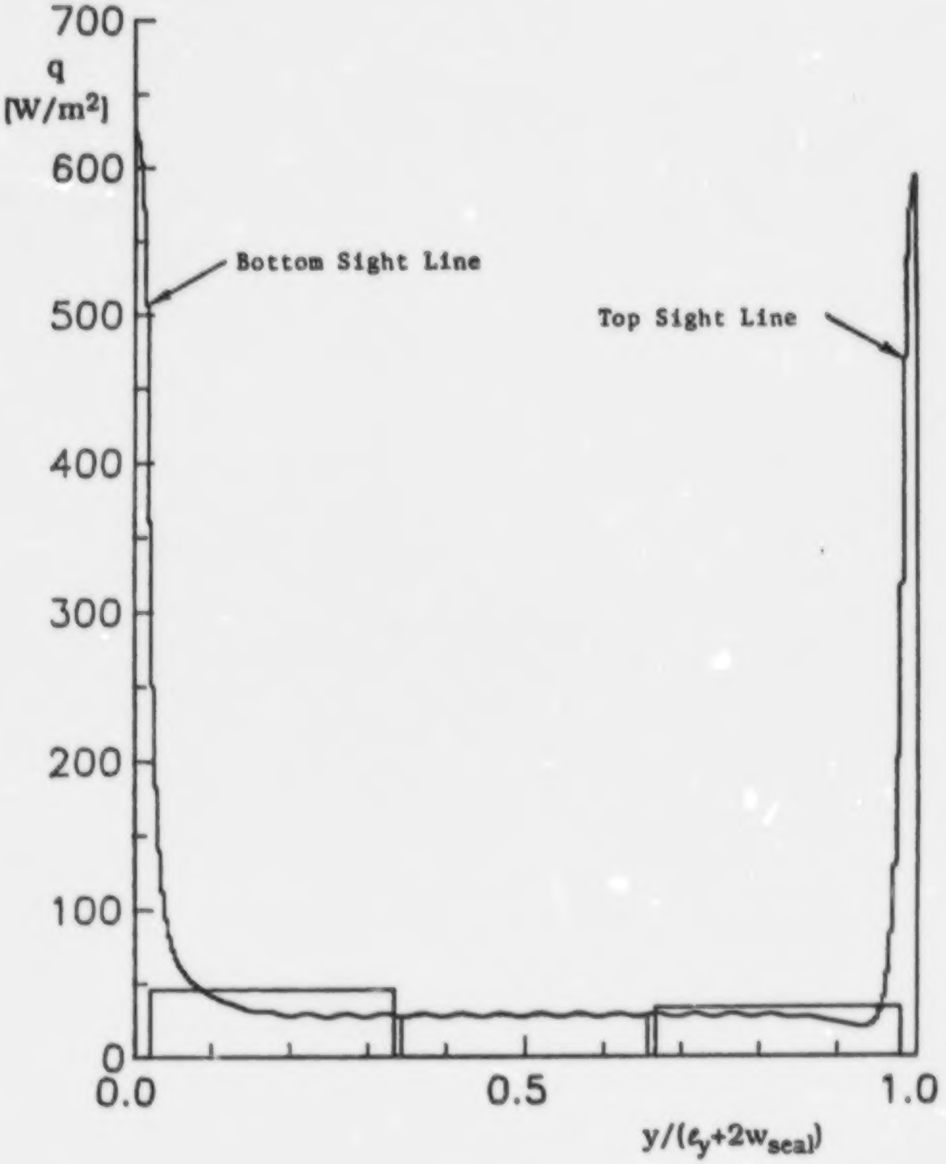


Figure 7.4: Calculated Heat Flux Distribution at Hot Boundary and Calculated Average Heat Flux for Heater Plate Areas (unit #3)

	CDG	DGLE	DGLE^{2}A	DGLEA	
Seals	cor'd	cor'd	conv'l	conv'l	
ℓ_x	12.1	12.4	14.1	12.0	[mm]
ΔT_{pp}	18.9	19.4	19.1	11.4	[K]
# cells	0	0	16	0	
Ra	3,355	4,145	7,490	2,623	
A	50.8	49.6	43.4	51.2	

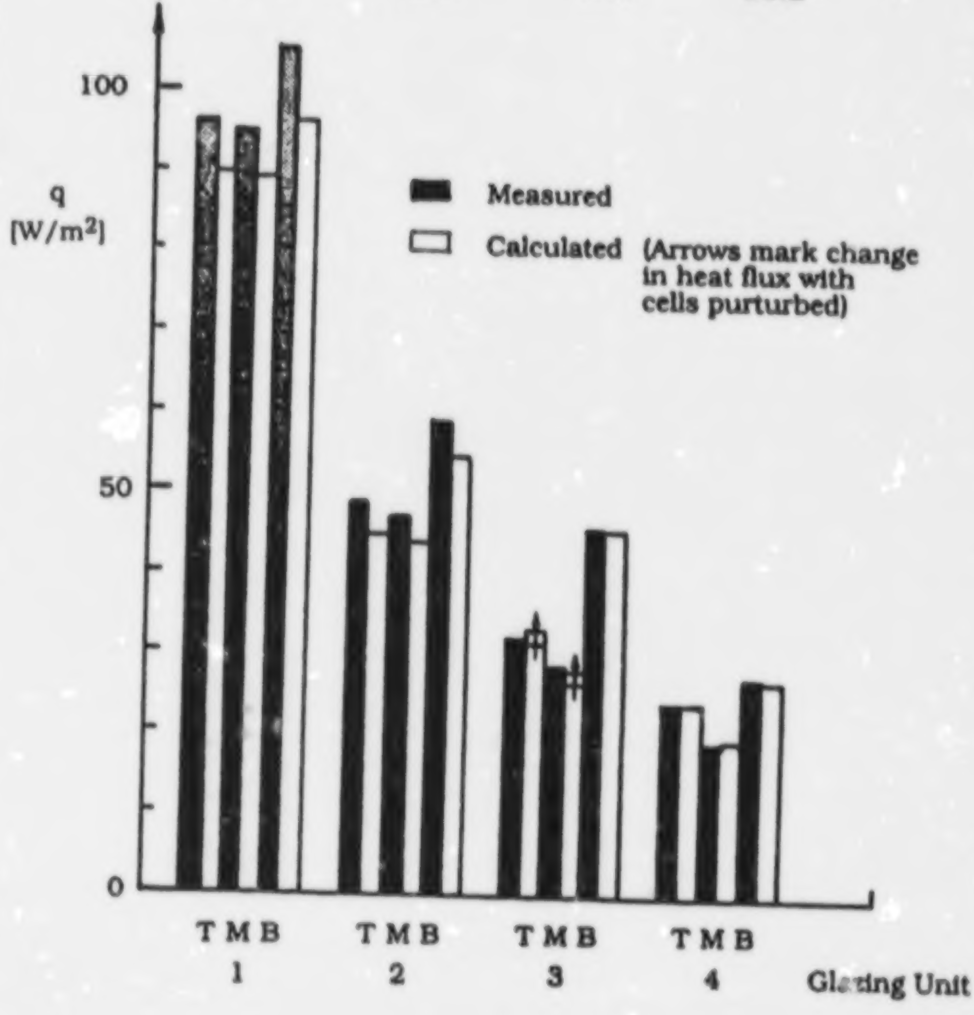


Figure 7.5a: Calculated and Measured Heat Flux Results for Top, Middle and Bottom Guarded Heater Plate Areas (units 1, 2, 3 and 4)

	DGLE	DGLE	DGLE	
Seals	Insulating Foam			
ℓ_x	16.0	13.0	13.0	[mm]
ΔT_{pp}	27.4	27.3	14.9	[K]
# cells	10	17	0	
Ra	15,955	8,437	4,638	
A	36.5	44.9	44.9	

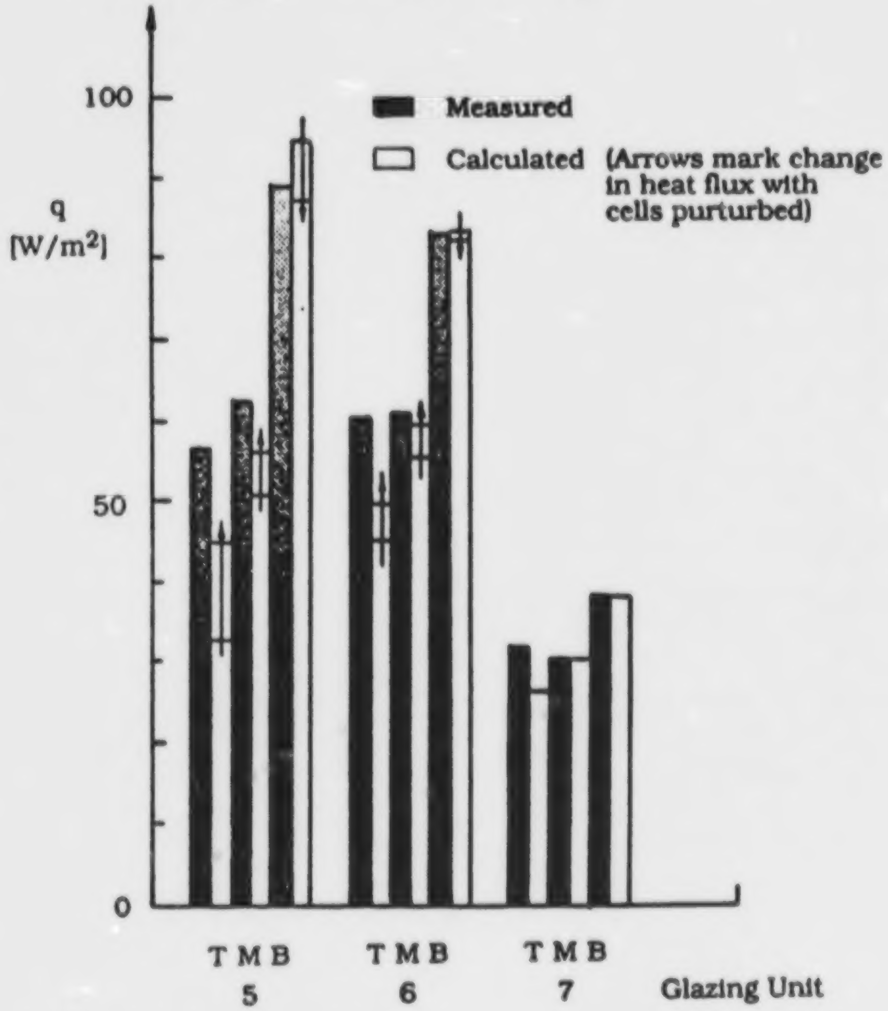


Figure 7.5b: Calculated and Measured Heat Flux Results for Top, Middle and Bottom Guarded Heater Plate Areas (units 5, 6 and 7)

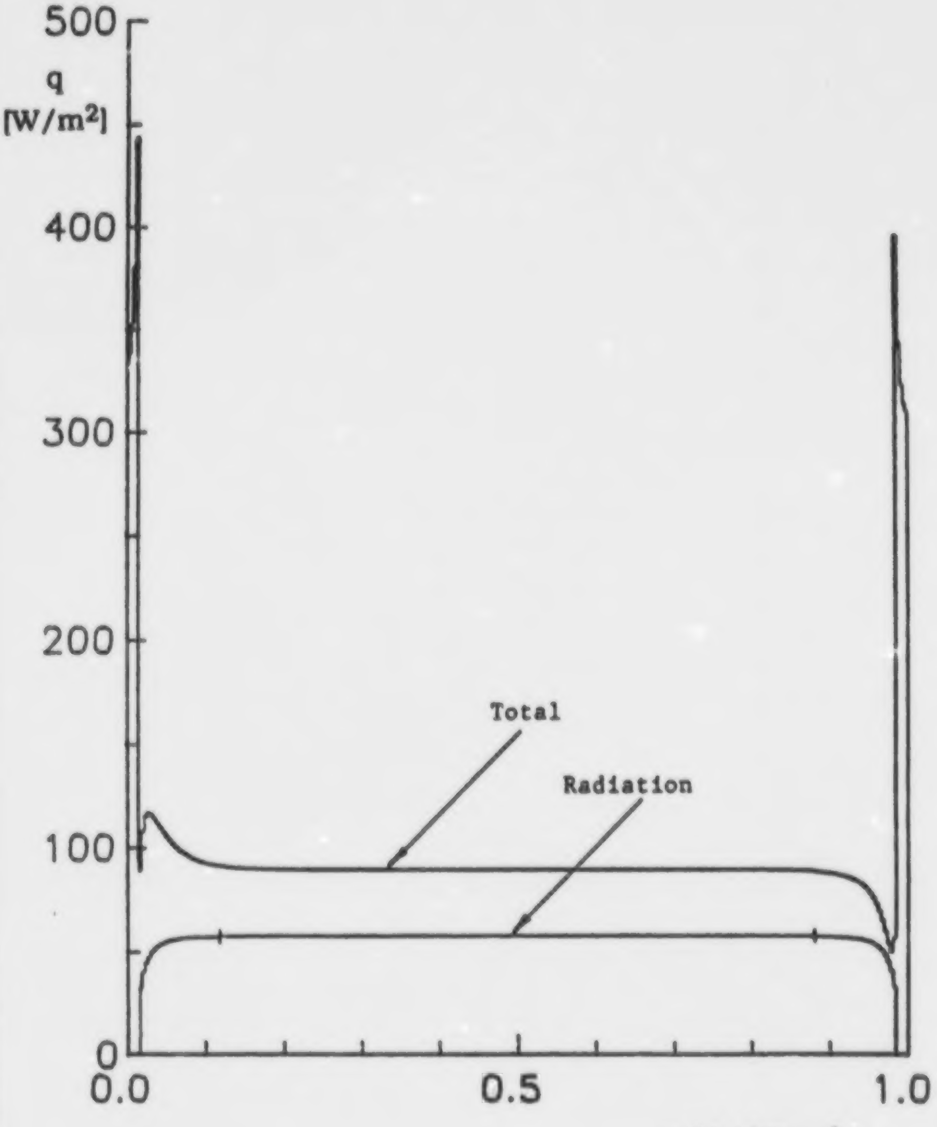


Figure 7.6: Heat Flux Component Distributions Along Vertical Cut at Hot Cavity Wall (unit #1)

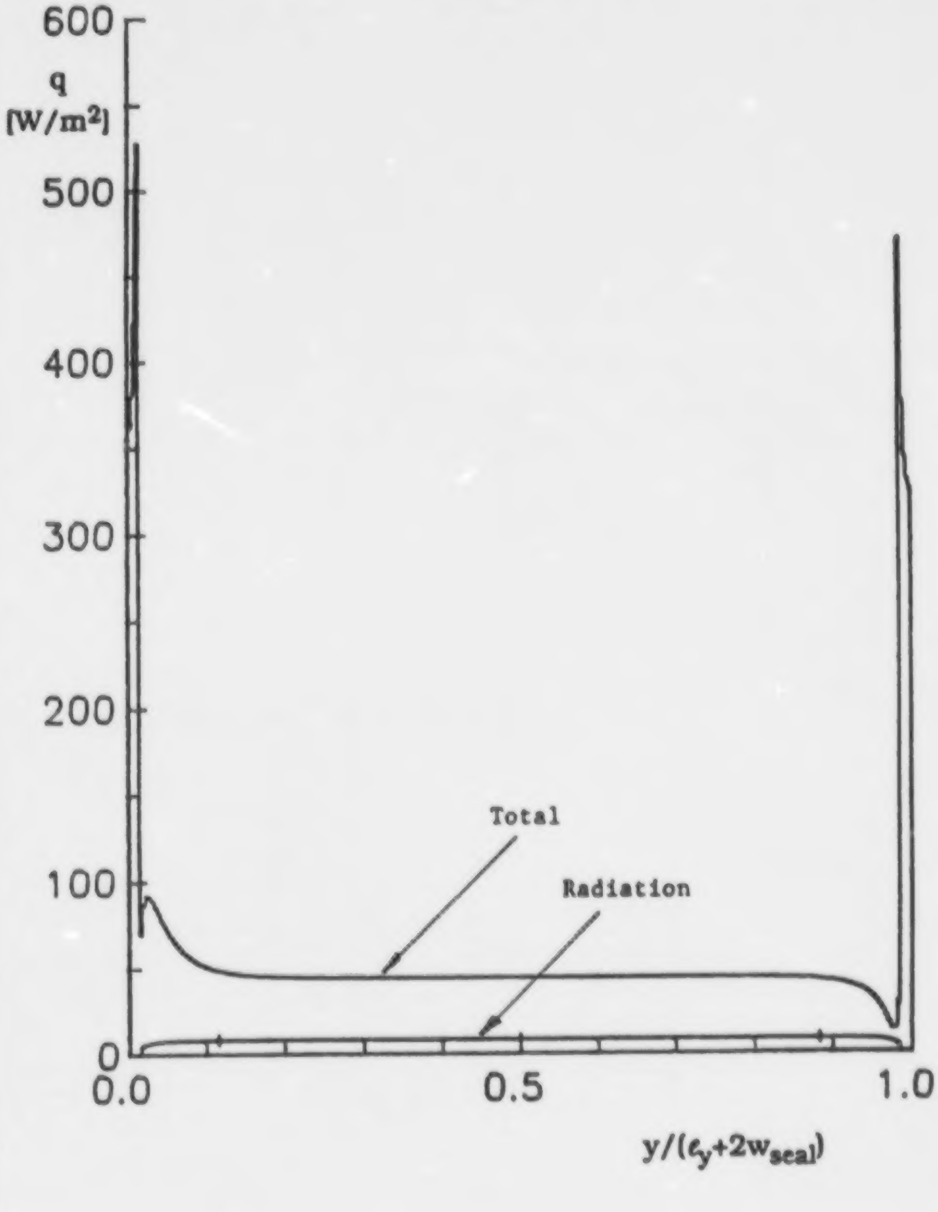


Figure 7.7: Heat Flux Component Distributions Along Vertical Cut at Hot Cavity Wall (unit #2)

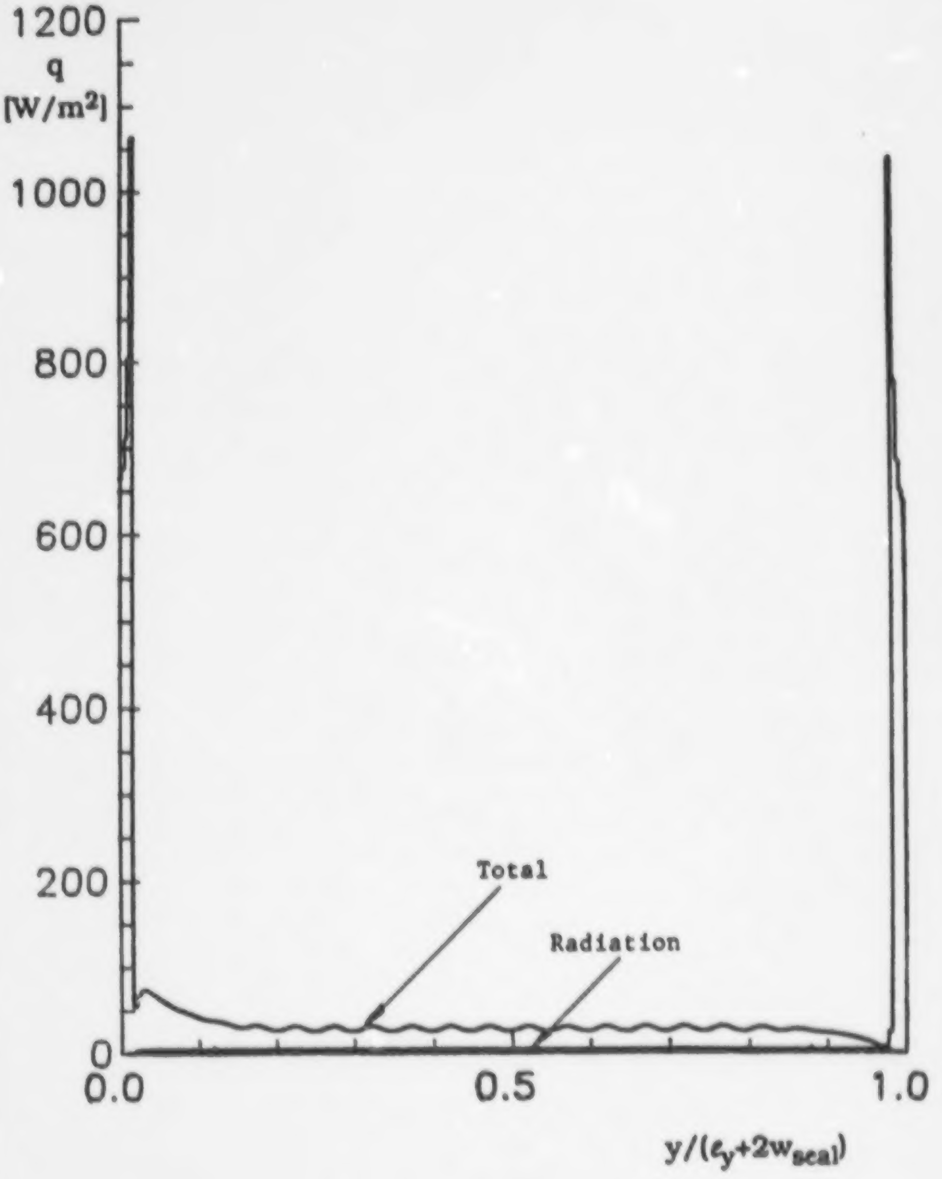


Figure 7.8: Heat Flux Component Distributions Along Vertical Cut at Hot Cavity Wall (unit #3)

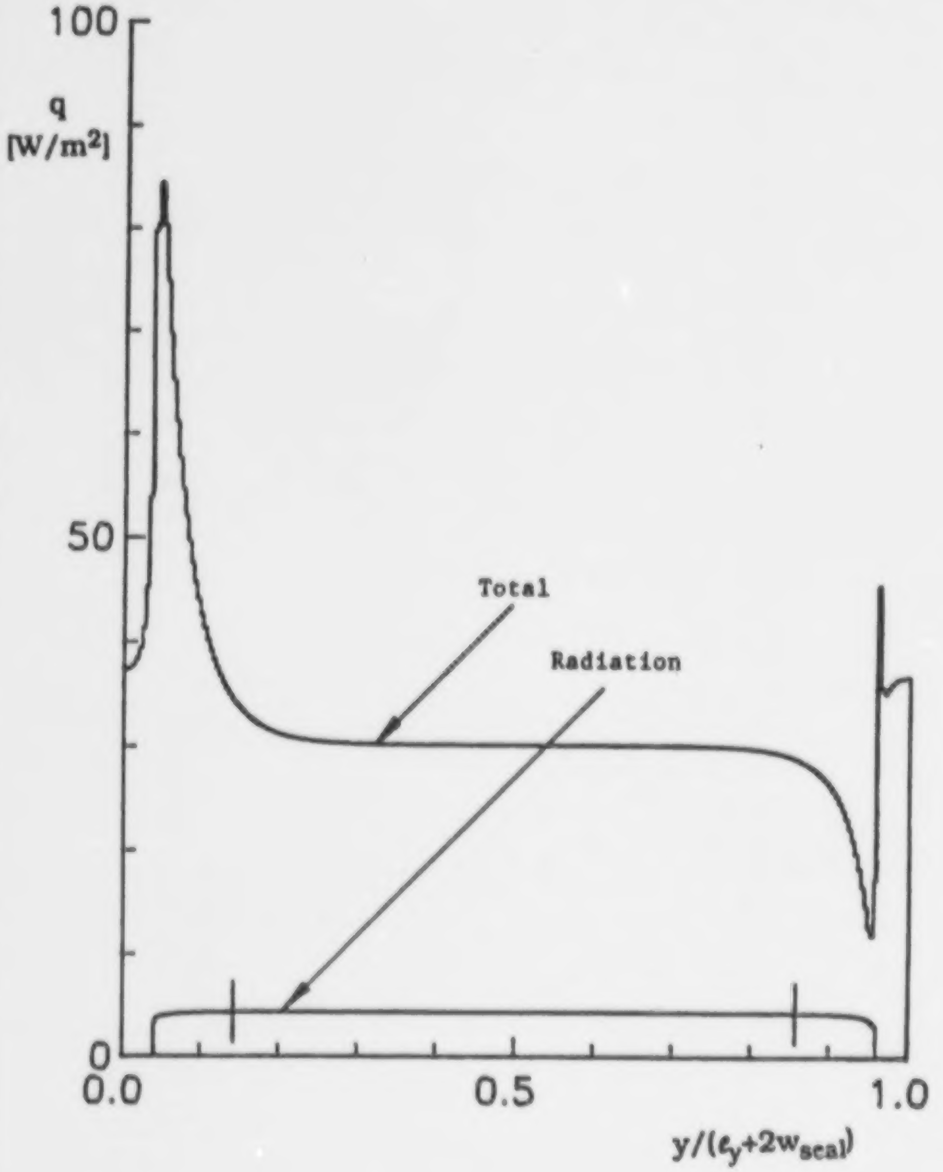


Figure 7.9: Heat Flux Component Distributions Along Vertical Cut at Hot Cavity Wall (unit #7)

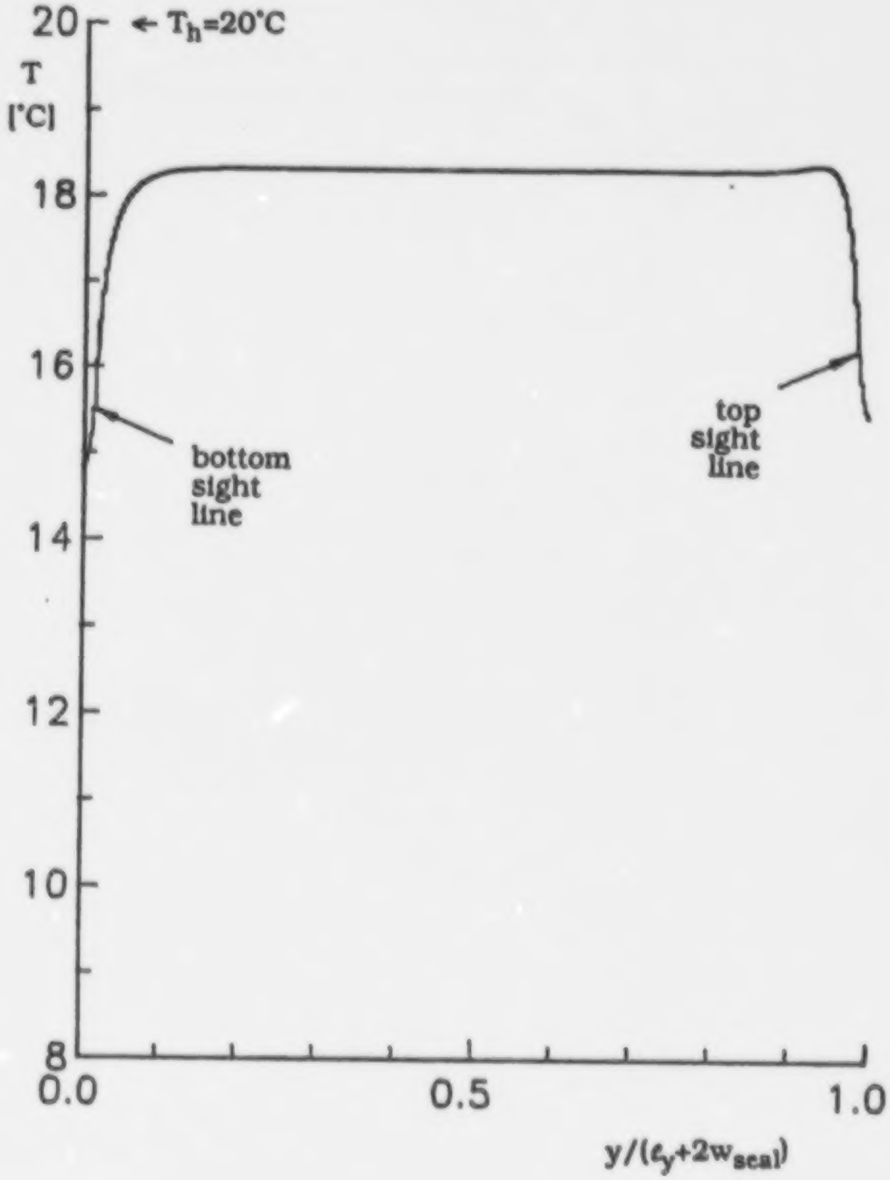


Figure 7.10: Temperature Distribution Along Vertical Cut at Hot Cavity Wall (unit #1)

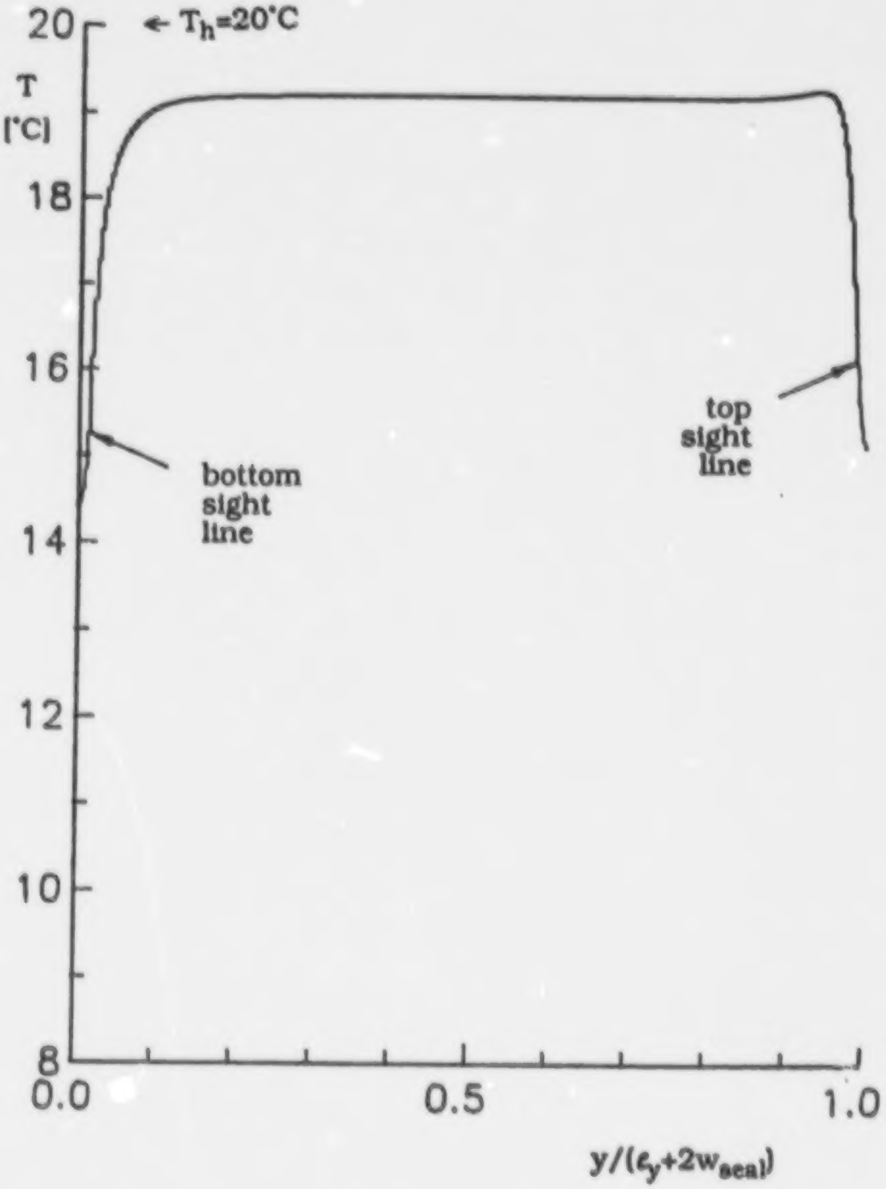


Figure 7.11: Temperature Distribution Along Vertical Cut at Hot Cavity Wall (unit #2)

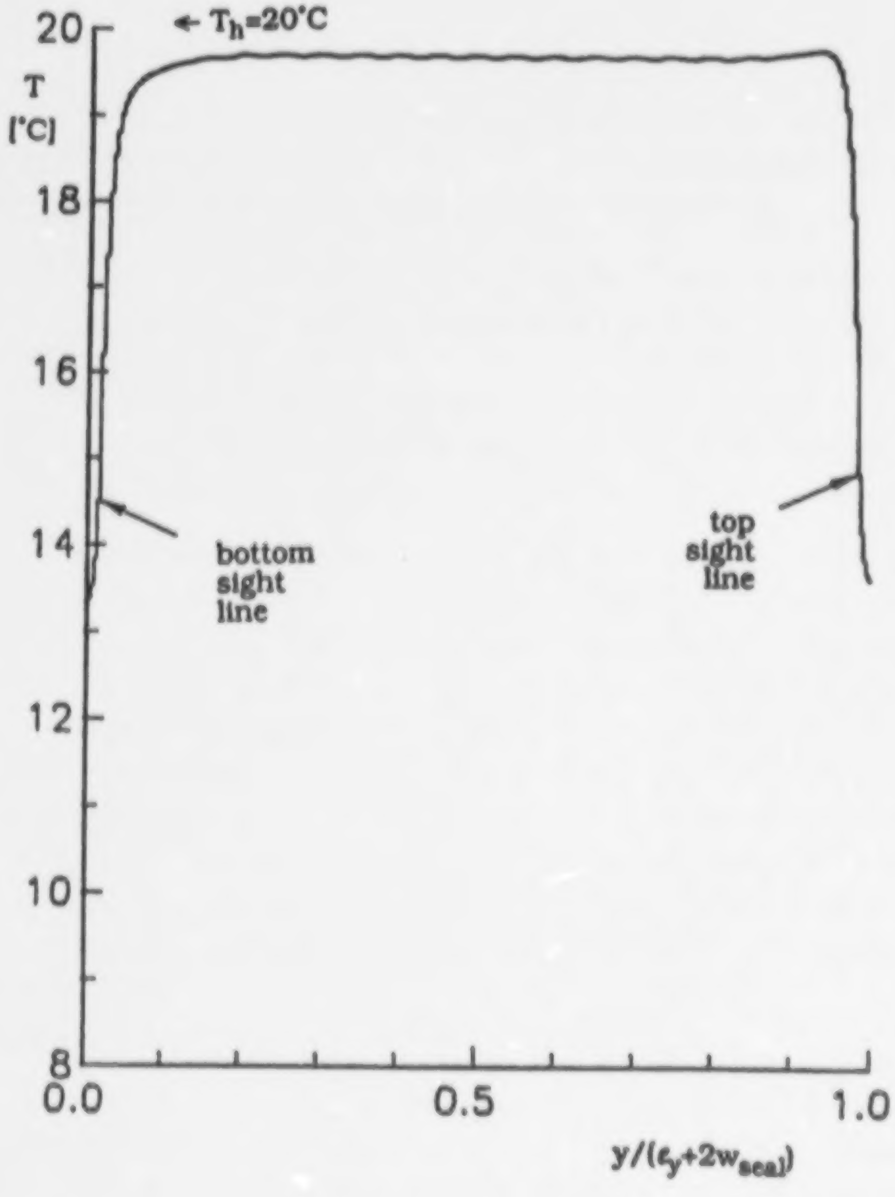


Figure 7.12: Temperature Distribution Along Vertical Cut at Hot Cavity Wall (unit #3)

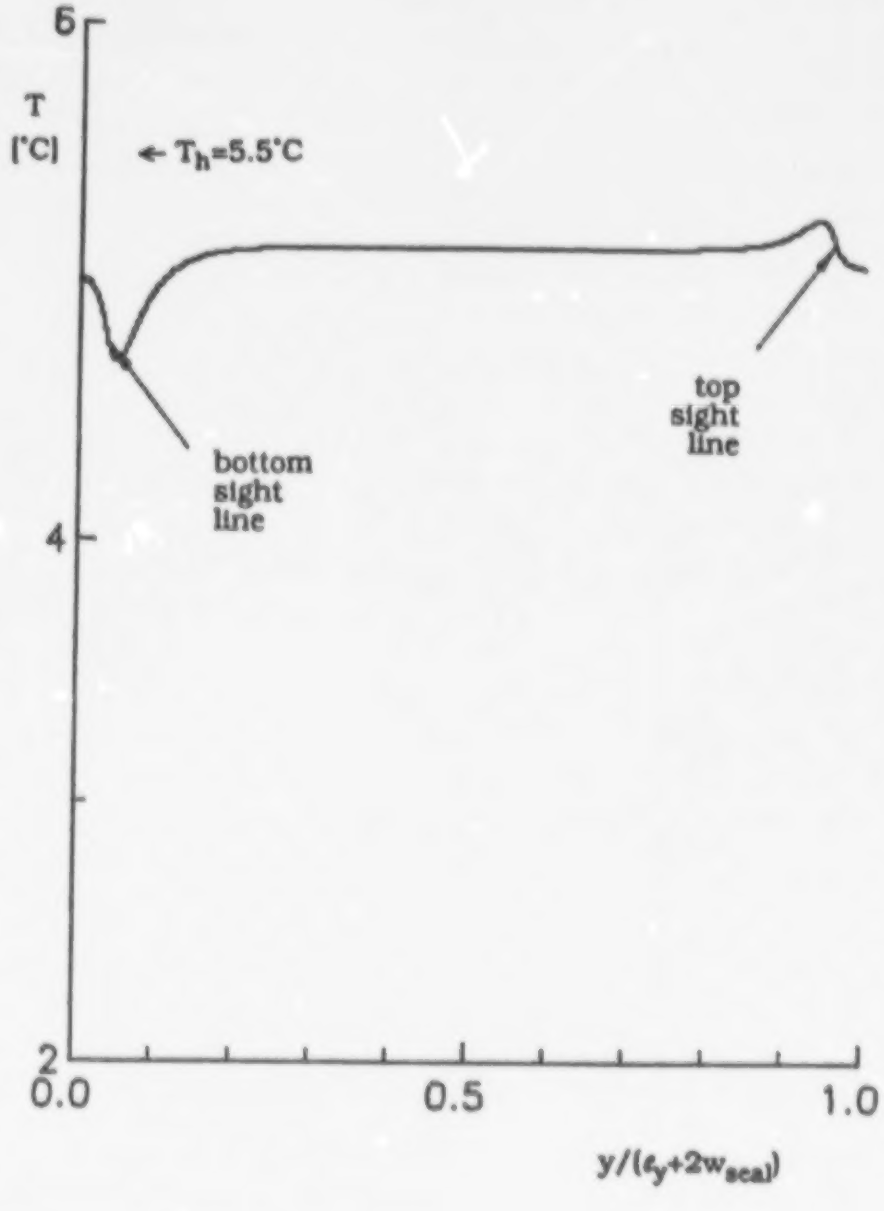


Figure 7.13: Temperature Distribution Along Vertical Cut at Hot Cavity Wall (unit #7)

Chapter 8

CONCLUSIONS

A variety of conclusions can be drawn based on the outcome of this research. Some are useful at a basic level and some concern specific problems currently being encountered in the window industry.

VISION has been enhanced in several ways including the provision of an optical data library, graphic output and the availability of additional fill gases. The U-value and shading coefficient calculations have been updated. In general, primarily on the basis of guarded heater plate test results, it can be concluded that VISION is an accurate design tool for the calculation of centre-glass heat transfer.

VISION provides capabilities not available in previous methods - most notably the ability to model glazings that are not opaque to longwave radiation. This capability has been highlighted by the study of Teflon films used as intermediate glazings adjacent to low emissivity surfaces. The advantages of using IR transparent films in conjunction with a selective surface in a solar collector were previously understood. In a comparable manner, this work has lead to the conclusion that highly IR transparent glazings can be used successfully in glazing systems. It has been demonstrated that Teflon film(s) placed adjacent to a low-e coating can attenuate radiative heat transfer in more than one interpane gap.

An experimental procedure using the UWNCA guarded heater plate apparatus for the measurement of heat transfer in glazing systems has been extended and refined. Consistent agreement betweeen measured and calculated centre-glass heat transfer rates have provided a high degree of confidence in the experimental results. Good agreement was also found when the results of 2-D simulations were compared with measured heat flux results for locations not within the centre-glass region. Similar edge-glass heat transfer measurements had not previously been undertaken. The low cost and simplicity of the guarded heat plate method have set it apart as a convenient and reliable development tool for prototype glazing systems. It can be concluded that a useful experimental glazing system design tool has been extended and demonstrated.

The guarded heater plate apparatus has also been adapted for the measurement of edge-seal thermal resistance. This procedure also constitutes a unique and useful developmental test capability. The experiment was developed to isolate and quantify the thermal performance of edge-seals with the specific purpose of allowing direct comparisons between various designs. No evidence of a similar experimental method exists in the literature. The test results recorded here comprise the only available experimental data regarding edge-seal thermal resistance. Many insights can be gained by examining these data. The most important observation is that the thermal resistance of edge-seals with aluminum spacer bars provide much less thermal resistance than any of the newer designs tested. It was also found that the placement of sealant can make a difference. Dual seal designs had less thermal resistance than similar single seal samples but this difference was only significant in units with aluminum spacer bars. The thermal conductivity of the sealant can influence the thermal resistance of the edge-seal – more appreciably in seals that do not have highly conductive spacer bars.

The study of natural convection in vertical, rectangular cavities has lead to a better understanding about fill gas flow and local heat transfer in glazing systems. At low values of Ra ($Ra<5,600$) a laminar, unicellular flow exists. At high values of Ra the flow will either become unsteady or it may develop a cats-eye pattern of steady, secondary cells. The cells can only exist for $A=12$ up to about $A=80$. A 2-D numerical model has been created to model the gas flow and heat transfer in the high aspect ratio vertical cavity. This model reproduces the results of previous researchers but involves a much simpler formulation. The results are deemed to be accurate on the basis of the extremely close agreement demonstrated between measured and calculated, local and average heat transfer rates. It can be concluded that this 2-D, numerical model accurately reproduces the physical mechanisms of natural convection in the vertical slot.

The natural convection model was combined with conduction and thermal radiation calculations in order to produce a 2-D model for heat transfer in glazing systems. Again the discrepancies between calculated heat fluxes and heat flux results from the guarded heater plate apparatus were small. It was found that it was not necessary to account for specular reflection of thermal radiation within the glazing unit cavity. No problems arose from the simplification of the radiation model in the centre-glass area. Heat flux and temperature profiles offered information regarding heat transfer in glazing systems. The validity of the ASHRAE definition of centre-glass and edge-glass areas was confirmed. The peak heat flux at the edge-seal can be more than an order of magnitude greater than the centre-glass heat flux if conventional edge-seals with aluminum spacer bars are used. The heat flux through spacers made of insulating foam does not differ greatly from the centre-glass heat flux but was still more conductive than the centre-glass for the samples tested. Simulation results repeatedly showed that the location of maximum heat flux to the warm glazing and the location of minimum temperature at the warm glazing is at the bottom sight line. This is where condensation most readily occurs. The minimum indoor glazing temperature is always found at the bottom sight line because of the motion of the fill gas and the resulting top-to-bottom asymmetry of the heat flux distribution. It can be concluded that IGU-2D accurately reproduces the mechanisms of heat transfer in glazing systems.

Extension of the 2-D glazing system model, IGU-2D, can now be pursued. Resulting models may be aimed at the quantification of heat transfer in glazing systems exposed to the environment, glazing systems coupled to window frames or for the prediction of minimum indoor pane temperatures and the presence of condensation. In this context, it can be concluded that an important outcome of the present research is the successful establishment of a proven glazing system model and the knowledge that accompanies this success. Future work can be undertaken with optimism.

Appendix A

Wright, J.L., "Effective U-Values and Shading Coefficients of Preheat/Supply Air Glazing Systems", Proceedings of the 1986 SESCI Conference, Winnipeg, Canada, pp. 219-224.

EFFECTIVE U-VALUES AND SHADING COEFFICIENTS OF PREHEAT/SUPPLY AIR GLAZING SYSTEMS

John L. Wright

Department of Mechanical Engineering
University of Waterloo
Waterloo, Ontario
N2L 3G1

ABSTRACT

Research is documented which makes use of a computer program called VISION. This computer program was developed specifically to provide a detailed analysis of heat transfer occuring in glazing systems. VISION was modified to perform an analysis of the energy flows in supply air windows. A model used to quantify heat transfer in the supply air flow is described. Ventilation air is brought in through supply air windows and the energy recovered by the preheat of the air flow is credited to the thermal performance of the window itself. Thus, the net energy flow between the conditioned space and the window was used to calculate an effective U-value and shading coefficient. The use of these "effective" window performance parameters permits the subsequent quantification of energy flows to or from the supply air glazing system without the necessity of modelling the detailed mechanisms of energy transport within the glazing system itself. A variety of glazing system designs are simulated. Indoor glazing temperature is reported for each system. In all cases the presence of preheat ventilation improved the effective shading coefficient moderately and increased the effective thermal resistance appreciably.

INTRODUCTION

Substantial energy conservation benefits are attainable through the use of "supply air windows". In order to reduce the space heating load the supply air required for ventilation passes into the building by way of a flow path provided within the glazing system instead of through leakage sites or openings located elsewhere in the building envelope. This glazing system arrangement is shown conceptually in Figure 1. Figure 1 also shows the numbering system that is applied to the indoor, outdoor, supply air and individual glazing temperature nodes. As the air passes between the glazings it is preheated by drawing on absorbed solar gains and/or thermal losses from the building interior. The need to restrict supply air flow solely to the window passages indicates that supply air windows should perform well when installed in a very airtight building envelope and used in conjunction with an exhaust air device that holds the building at a slightly negative pressure.

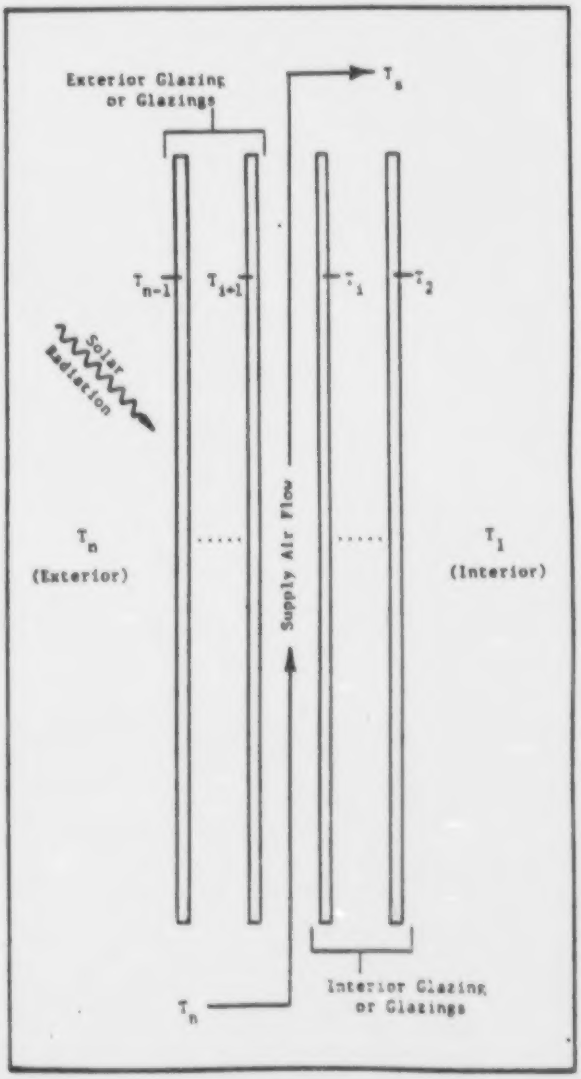


Figure 1: Preheat/Supply Air Flow Concept

Results of the relatively modest amount of research that has been performed on supply air windows are encouraging. Korkala, etal [1] examined some practical aspects of several double glazed configurations showing that useful amounts of supply air could be taken in without serious loss of thermal comfort. Barakat has noted [2] a decrease in required space heating energy of about 25% when one unit of the NRC Passive Solar Facility [3] (located in Ottawa) was fitted with triple glazed supply air windows and compared to a similar unit fitted with conventional double glazed windows. A decrease in required energy was also noted when triple glazed supply air windows were compared to conventional triple glazed windows.

The VISION computer program [4] has been modified in order to simulate the performance of supply air glazing systems. VISION was developed specifically to provide reliable U-value and shading coefficient estimates for innovative glazing systems. Heat transfer within the glazing system is quantified using a detailed one dimensional (1-D) analysis. The capabilities of this analysis have been extended to include the treatment of supply air flow between glazings without resorting to a major rewrite of the VISION source code. Instead, the existing program has been modified to only a limited degree and is tricked to model the internal air flow. The conduction/convection heat transfer rates from the walls adjacent the forced air flow are determined using a two dimensional (2-D) analytical solution [5] of basic equations describing the elemental energy balance in the fluid flow. The two solutions that are applied - one dealing with heat transfer between glazings and the other with the heat transfer to the forced flow of supply air - are well suited to be applied simultaneously since both are based on the assumption that each glazing is isothermal.

The results of this analysis serve not only to support the intuitive and qualitative understanding of preheat/supply air glazing systems. In addition they provide quantitative feedback regarding the merits of various design options, their energy conservation potential and the principles governing their operation. For instance, it is recognized that the flow of supply air between the glazings will lower the temperature of the indoor glazing. Indoor glazing temperature is reported for each system simulated in order to determine the change in allowable indoor humidity.

Glazing system designs range from a simple double glazed system to more complex configurations involving the use of two, three or four glazings, low emissivity (low E) coatings and argon fill gas.

THE SUPPLY AIR FLOW MODEL

The following paragraphs present an overview of the technique used to model the heat transfer between the glazing system and the supply air flow. This presentation is not complete in every detail. The remaining information can be found in the references provided. The nomenclature needed to understand this analysis is shown in Figure 2.

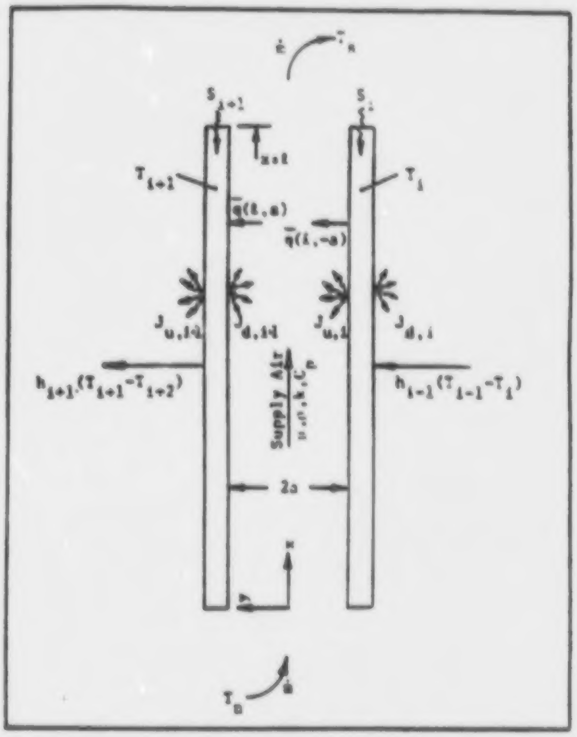


Figure 2: The Supply Air Flow Model

It can be shown that the Reynolds number based on hydraulic diameter and average air speed, Re_D, of the supply air flow is given by:

$$Re_D = 2\dot{m}\ell/\mu \qquad (1)$$

where $\dot{m}$ is the mass flow rate per unit aperture area, μ the dynamic viscosity of the air and the length of the flow path is ℓ. ℓ is equal to the height of the window. Note that the hydraulic diameter, D, is equal to four times the half width of the flow gap.

$$D = 4a \qquad (2)$$

The value of $\dot{m}$ can be calculated using equation 3.

$$\dot{m} = \rho H(A_f/A_a)ACR \qquad (3)$$

where ρ is the air density, H is ceiling height, ACR is the air change rate and A_f/A_a is the ratio of floor area to aperture area. Based on the following assumed values,

$H = 2.44$ m

ACR = 0.5 air changes per hour

and $A_f/A_a = 10$

the value of $\dot{m}$ was taken to be 15.6 Kg m^{-2} hr^{-1} for all simulations carried out. Further assuming that ℓ=1m the Reynolds number was estimated to be Re_D=525.

The Reynolds number result shown above indicates that the supply air flow can safely be assumed to be laminar in nature. Referring to equation 1, it can be seen that the height of the window can approach 4m before the critical Reynolds number marking the onset of turbulence (approx. 2000) is approached. The supply air flow in all simulations is treated as laminar, steady and fully developed.

An energy balance applied to an elemental control volume in the supply air flow, while neglecting conduction in the flow direction, yields equation 4:

$$u(\partial T/\partial x) = a(\partial^2 T/\partial y^2) \qquad (4)$$

where a is the thermal diffusivity of the air and u is the velocity component in the x direction. Solutions to equation 4 can be found in references 5 through 10 among others. The particular solution used in this analysis is found in reference 5 where Cess and Shaffer express the 2-D temperature solution in terms of Stieltjes integrals. Application of the Fourier-Biot law to the temperature field yields an expression for the local heat flux in the y direction.

$$q(x,y) = (-k/a)[(1 - 2\theta'(x,y))(T_{i+1}-T_n) + (-1 + 2\theta'(x,-y))(T_i-T_n)]/2 \qquad (5)$$

Once the flow space half width, a, and the Peclet number are fixed the function $\theta'(x,y)$ is a function of x and y only. The Peclet number, Pe, is given by:

$$Pe = Re_D\ Pr \qquad (6)$$

where Pr is the Prandtl number of the air flow. An expression for $\theta(x,y)$ is given in reference 5.

The average heat flux to the supply air at the adjacent glazings can be found by setting y=a or y=-a in equation 5 and integrating with respect to x from x=0 to x=ℓ.

$$\bar{q}(\ell,a) = (1/\ell)\int_0^{\ell} q(x,a)\ dx \qquad (7)$$

$$\bar{q}(\ell,-a) = (1/\ell)\int_0^{\ell} q(x,-a)\ dx \qquad (8)$$

These average heat transfer rates, $\bar{q}(\ell,a)$ and $\bar{q}(\ell,-a)$ are shown in Figure 2. The net heat flux from the glazings to the supply air is given by:

$$q_s = \bar{q}(\ell,-a) - \bar{q}(\ell,a) \qquad (9)$$

and can be used to calculate the temperature of the supply air as it leaves the glazing system, T_s.

$$T_s = T_n + q_s/(\dot{m}C_p) \qquad (10)$$

where C_p is the specific heat of the supply air.

INCORPORATION OF THE SUPPLY AIR MODEL

The VISION computer program quantifies the various energy flows within a glazing system by simultaneously applying an energy balance to each individual glazing. Reference 4 can be used as a complete source of detailed information. This energy balance applied at the i^{th} glazing is given by equation 11.

$$h_{i-1}(T_{i-1} - T_i) + J_{u,i-1} - J_{d,i} + S_i = h_i(T_i - T_{i+1}) + J_{u,i} - J_{d,i+1} \qquad (11)$$

The heat transfer coefficients, h_{i-1} and h_i, are used to quantify convective heat transfer from glazing i-1 to glazing i and from glazing i to glazing i+1, respectively. The quantities J_u and J_d represent radiosities that account for various upward and downward (in this case outdoor and indoor) bound fluxes of thermal radiation. Solar radiation which is absorbed at the i^{th} glazing and appears as an energy source is equal to S_i.

Referring to the i^{th} glazing shown in Figure 2 it can be seen that equation 11 cannot be directly applied. The heat transfer coefficient h_i has no apparent meaning in the supply air gap. Instead, the VISION computer code is tricked using a two step procedure. First, the coefficient h_i at the air flow gap is set equal to zero. This eliminates what VISION sees as convective heat transfer across the gap. Second, VISION is supplied with a modified source term, S_i', in the place of S_i. S_i' represents the energy source due to absorbed solar radiation minus the energy lost to the supply air flow.

$$S_i' = S_i - \bar{q}(\ell,-a) \qquad (12)$$

Substitution into equation 11 yields a more meaningful energy balance.

$$h_{i-1}(T_{i-1} - T_i) + J_{u,i-1} - J_{d,i} + S_i' = J_{u,i} - J_{d,i+1} \qquad (13)$$

Applying the same procedure at glazing i+1 gives:

$$S'_{i+1} = S_{i+1} + \bar{q}(i,a) \qquad (14)$$

and

$$J_{u,i} - J_{d,i+1} + S'_{i+1} = h_{i+1}(T_{i+1} - T_{i+2}) + J_{u,i+1} - J_{d,i+2} \qquad (15)$$

VISION incorporates an iterative procedure where glazing temperatures are determined as a function of the various heat transfer coefficients within the glazing system/-environment array. At each iteration the heat transfer coefficients are updated and iteration continues until the temperature solution converges. In the supply air solution the heat flux to the supply air was also updated at each iteration. When convergence was reached a full solution including all glazing temperatures, heat transfer rates and supply air temperatures had been found.

CALCULATION OF EFFECTIVE U-VALUE

The U-value is calculated using an indoor temperature of T_i=21 C, an outdoor temperature of T_n = -18 C, a fraction cloud cover of 0.5 and zero solar radiation. See reference 4 for more detail. VISION performs this calculation using the total heat flux from the conditioned space to the interior glazing. An effective U-value, U_e , can be calculated by crediting the window with the energy gained by the supply air flow, q_s.

$$U_e = [h_1(T_1 - T_2) + J_{u,1} - J_{d,2} - q_s]/(T_1 - T_n) \qquad (16)$$

In the absense of solar radiation the heat flux from the interior to the inside glazing is greater than the heat flux from the outside glazing to the exterior environment by the amount of energy transferred to the supply air, q_s. Therefore, a simpler expression for the effective U-value is:

$$U_e = [h_{n-1}(T_{n-1} - T_n) + J_{u,n-1} - J_{d,n}]/(T_1 - T_n) \qquad (17)$$

CALCULATION OF EFFECTIVE SHADING COEFFICIENT

The shading coefficient is calculated in the absence of indoor/outdoor temperature difference (T_i=T_n=19 C), 741 W m^{-2} of incident solar radiation and zero cloud cover. Reference 4 provides detailed information. The effective shading coefficient , SC_e , is found when the window is credited with the supply air energy gain, q_s, in addition to the customary quantities of absorbed/retransmitted and directly transmitted solar gain. The total energy gain is divided by the energy gain of a conventional single glazed window subjected to the same conditions , q_{ref}.

$$SC_e = [h_1(T_2 - T_1) + S_1 + q_s]/q_{ref} \qquad (q_{ref} = 612.3 \text{ W m}^{-2}) \qquad (18)$$

OPTICAL PROPERTIES

All of the glazing systems simulated consisted of conventional glass with or without one of two types of low E coating. Input data used to characterize the solar properties of glass were index of refraction (n=1.52) and extinction coefficient (K=0.024 mm^{-1}). All glazings were taken to be 3.175 mm thick. The glass/low E glazing element was modelled as having 60% normal solar transmissivity, and normal reflectivities of 25% and 22% with respect to solar radiation incident from the coating or substrate sides, respectively.

The hemispheric/hemispheric long wave (λ >3µm) properties were specified such that all glazings were opaque. The emissivities of the low E coating used in sealed spaces and of plain glass were 0.09 and 0.84, respectively. In cases where a low E coating was used in an air flow gap the emissivity was set at 0.35 in order to model the more durable coating that would be required.

SIMULATION RESULTS

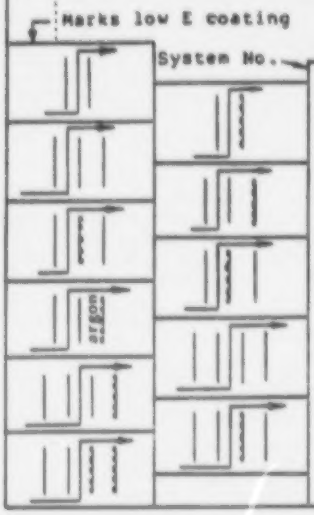
Computer simulations were carried out for eleven different glazing systems that incorporate a variety of low E coating(s), air flow locations and fill gas options. All pane spacings including the supply air flow gap were set at 15 mm. Each glazing system was simulated with and without supply air flow. Thus, four primary results consisting of two U-values and two shading coefficients were calculated for each system. These simulation results plus a variety of intermediate calculation results are presented in Table 1.

Examining the U-value results shown in the left hand portion of Table 1 a number of observations can be made. In all of the glazing systems simulated the presence of supply air flow reduced the U-value. Although the amount of reduction in U-value might vary widely from one system to another, it is noteworthy that this reduction was always appreciable. The smallest U-value reduction was 31%.

It is also remarkable that the U-value reduction can be greater for systems that are more intricate and that have relatively low U-values without supply air flow. Most of the two and three glazing systems exhibit a 30 to 40% U-value reduction but all of the four glazing systems provide a reduction of more than 50% when supply air is used. This sizeable reduction of an already low U-value results in exceptionally low effective U-values. For instance, system eleven shows a 58% reduction in U-value giving an effective U-

TABLE 1: PREHEAT/SUPPLY AIR SIMULATION RESULTS

Subscript "e" indicates presence of supply air



System No.	U-VALUE					SHADING COEFFICIENT			
	U W/m^{2}C	T_2 C	$T_{2,e}$ C	T_s C	U_e W/m^{2}C	S_1 W/m^2	SC	q_s W/m^2	SC_e
1	2.88	6.2	4.3	-6.7	1.94	501.2	0.87	10.4	0.88
2	2.35	9.1	7.3	-5.7	1.30	358.6	0.71	21.0	0.72
3	1.84	11.8	10.9	-11.0	1.22	425.4	0.78	19.8	0.80
4	1.32	14.5	13.9	-13.1	0.88	307.3	0.65	26.5	0.68
5	1.32	14.5	13.9	-13.1	0.88	306.9	0.60	41.8	0.65
6	1.59	13.1	12.1	-9.9	0.87	306.9	0.63	36.7	0.67
7	1.12	15.5	14.9	-13.5	0.76	307.3	0.65	26.4	0.68
8	1.34	14.4	11.7	-7.5	0.67	362.5	0.71	40.6	0.74
9	1.01	16.1	14.4	-10.5	0.49	263.7	0.59	53.3	0.65
10	1.19	15.2	12.5	-7.4	0.51	263.6	0.59	58.1	0.63
11	0.90	16.6	14.9	-10.3	0.38	194.8	0.49	77.2	0.57

value of 0.38 W m^{-2} C^{-1}. This is equivalent to a thermal resistance of RSI 2.6 or R 14.9.

The results of the U-value calculations also highlight a relationship between the location of the supply air flow (or the location of thermal resistance) and supply air temperature, T_s. As more thermal resistance is added between the supply air flow and the conditioned space the temperature of the supply air flow drops. Compare system one and system three. The addition of one glazing on the interior decreases effective U-value by 37% but at the same time the supply air temperature drops from -6.7 C to -11.0 C. A similar comparison can be made between systems three and four where the only change is the addition of one low E coating. On the other hand the addition of thermal resistance between the supply air gap and the exterior environment not only decreases the effective U-value but an increase in supply air temperature is produced at the same time. This effect can be seen by comparing systems three and eight, four and nine or six and ten.

The inside glazing temperature also varies according to the presence of supply air flow and as a function of the placement of thermal resistance. The presence of supply air always reduces the temperature of the inside glazing. If thermal resistance is added anywhere in the glazing system the temperature of the inner glazing rises and approaches the temperature of the conditioned space. Compare systems three and four, three and six as well as three and eight. However, it is worth noting that only in the case of system one itself was the inside glazing temperature - with supply air present - lower than the inside glazing temperature of a conventional double glazed system (system one) operated without supply air flow. The inside glazing temperature of system three with supply air flow was 10.9 C. This surface temperature corresponds to an allowable relative humidity level in excess of 50%.

The use of a low E coating in the supply air flow gap illustrates an interesting phenomenon. Compare systems five and six. These two configurations provide essentially the same effective U-value but system six delivers supply air at -9.9 C rather than -13.1 C. The same effect can be seen by comparing systems nine and ten. A low E coating placed in the air flow gap suppresses overall heat transfer by reducing radiative exchange across this gap but does not act to reduce the energy transferred directly from the interior to the supply air. In contrast, a low E coating placed in the interior side sealed space does reduce energy flow from the interior to the supply air causing the supply air to enter the room at a lower temperature. As a result of this effect, a low E coating placed in the flow air gap, rather than in the interior side sealed gap, appeared to be of greater benefit. Not only were the resulting supply air temperatures higher but the effective U-values were comparable in spite of the higher emissivity of the durable low E coating used in the flow air gap.

The shading coefficient calculation results shown on the right hand side of Table 1. also show a number of interesting trends. In all systems simulated the presence of supply air flow increased the shading coefficient. Most of these increases are marginal - in the range of from 1 to 5%. Similar to the change in U-value the increase in shading coefficient was greater, as a general trend, for more intricate glazing systems. For example, the shading coefficient of system eleven which has four panes and two low E coatings increased by 16% when supply air was introduced. On the other hand, the more intricate glazing systems have lower shading coefficients and effective shading coefficients as a general rule.

The effective shading coefficient appears to be relatively insensitive to the location of low E coatings that are placed between the supply air flow and the interior. Compare systems four to system five or six. In system four the solar radiation that is absorbed at the low E coating can readily transfer to the conditioned space directly. In systems five and six the solar radiation absorbed at the coating more readily transfers to the supply air and is again supplied to the interior. In the latter case the supply air is given more preheat energy.

CONCLUDING REMARKS

A calculation method has been presented which makes possible the analysis of a preheat/supply air flow passage within the framework of the VISION computer program. Results are consistent with intuition regarding the mechanisms governing operation of this type of window. Reliable generalizations or "rules of thumb" are difficult to make when dealing with the design of supply air windows. The number of design options is extremely large. Where particular design options are considered a detailed simulation of each system is recommended.

It is noted that the introduction of supply air flow consistently decreased the U-value and increased the shading coefficient of the various glazing systems simulated. These changes were generally larger for more intricate glazing designs. The effective U-values of four glazing systems can be exceptionally small without seriously sacrificing solar gains.

The use of multiple panes and low E coatings can be used to tailor effective U-value and effective shading coefficient in much the same way as is possible with conventional glazing systems. Similar trade-offs exist. In addition, there exists a trade-off between increasing the temperature of the supply air flow and lowering the temperature of the interior glazing. Results from the simulations carried out indicate that interior glazing temperatures are not low enough to aggravate water condensation problems in all but the very simple double glazed system. In light of this finding, it is expected that glazing system design can be more directly determined by the possibility of maximizing the temperature of the supply air. In order to accomplish this goal the use of a durable low E coating in the air flow gap looks promising and warrants further attention.

Effective U-values and effective shading coefficients can be used directly in the place of U-values and shading coefficients as input to building energy analysis programs. However, caution must be exercised because these results are based on a specific rate of ventilation air flow inward (per unit aperture area) at the building envelope. This same inflow rate of unheated air must be included in the building energy analysis. The superior performance of the supply air/preheat window relies on the fact that this ventilation air would be required regardless of its infiltration path.

The work presented in this study raises many questions. It is intended that future work will include prototype testing, the simulation of multiple pass supply air flow passages as well as the sensitivity analysis of performance results with respect to air flow rate and the length and thickness of the air flow gap.

REFERENCES

[1] Horkala, T., Saarnio, P., Siitonen, V., "Air Intake Arrangements of the Supply Air Window from the View of Comfort and Ventilation Efficiency", Proceedings of the Windows in Buildings Conference, Sweden, June, 1984

[2] Barakat, S.A., Personal communication, December, 1985

[3] Barakat, S.A., "NRCC Passive Solar Test Facility Description and Data Reduction", DBR/NRC Technical note BRN #214, May, 1984.

[4] Ferguson, J.E., Wright, J.L. "VISION: A Computer Program to Evaluate the Thermal Performance of Super Windows", Solar Energy Program, NRC report no. Passive-10.June, 1984.

[5] Cess, R.D., Shaffer, E.C., "Summary of Laminar Heat Transfer Between Parallel Plates With Unsymmetrical Wall Temperatures", Journal of the Aero/Space Sciences, August, 1959, page 538.

[6] Lundberg, R.E., Reynolds, W.C., Kayes, W.M., "Heat Transfer with Laminar Flow in Concentric Annuli with Constant and Variable Wall Temperature with Heat Flux", NASA TN D-1972, Washington, August, 1963.

[7] Yih, C.S., Cermack, J.E., "Laminar Heat Convection in Pipes and Ducts", Civel Engineering Department, Colorado Agricultural and Mechanical College, Fort Collins, Colorado, September, 1951, ONR Contract No. N90nr-82401, NR 063-071/1-19-49.

[8] Prins, J.A., Mulder, J., Schenk, J., "Heat Transfer in Laminary Flow Between Parallel Plates", Applied Scientific Research, Section A Vol 2, pp. 431-438, 1949-51.

[9] Schenk, J., Beckers, H...., "Heat Transfer in Laminar Flow Between Parallel Plates", Applied Scientific Research, Section A, Vol 4, pp. 405-413, 1953-54.

[10] Sellars, J.R., Tribus, M., Klein, J.S., "Heat Transfer in Laminar Flow in a Round Tube or Flat Conduit - The Graetz Problem Extended, Trans. ASME, Vol. 78, pp. 441-448, 1956.

ACKNOWLEDGEMENTS

The author acknowledges financial support from The Renewable Energy Branch of The Department of Energy, Mines and Resources.

Appendix B

Sample IGU-2D Input File; Glazing System No. 3, Table 7.1

```
IGU3, Cardinal unit #1, tested Oct. 5, 1989, 3.0/14.0/3.0E2
klin = 2.0 W/m.C
May 25, 1989
2               | ib (integer)
30              | ie (integer)
2               | jb (integer)
7               | jbc (integer)
0.88            | tww, west temperature of problem domain, C, (real)
20.0            | twe, east temperature of problem domain, C, (real)
.01412          | lx, length of IGU cavity in x direction, m, (real)
.6129           | ly, length of IGU cavity in y direction, m, (real)
101.3           | pb, pressure of gas in IGU cavity, kPa, (real)
2               | idgas, fill gas identifier, (integer),  eg. air=1
1.              | mfrac(1), mole fraction of 1st gas component, (real)
2.56            | kseal, thermal conductivity of seal, W/mK (real)
0.01105         | wseal, width of seal in contact with glass, m (real)
0.96            | kg, thermal conductivity of glass, W/mK (real)
0.0033          | tg, thickness of glass, m (real)
0.17            | kn, thermal conductivity of neoprene, W/mK (real)
0.001524        | tn, thickness of neoprene mats, m, (real)
10              | nitert, max. no ADI iterations used for T (integer)
4               | niteru, max. no ADI iterations used for u(integer)
5               | niterv, max. no ADI iterations used for v (integer)
7               | niterp, max. no ADI iterations used for pp (integer)
10000.          | tolt, residual reduction factor for T (real)
10000.          | tolu, residual reduction factor for U (real)
10000.          | tolv, residual reduction factor for v (real)
10000.          | tolp, residual reduction factor for pp (real)
0               | iblkit, block correction flag for T, i dir'n, (integer)
0               | iblkjt, block correction flag for T, j dir'n, (integer)
0               | iblkiu, block correction flag for u, i dir'n, (integer)
0               | iblkju, block correction flag for u, j dir'n, (integer)
0               | iblkiv, block correction flag for v, i dir'n, (integer)
0               | iblkjv, block correction flag for v, j dir'n, (integer)
0               | iblkip, block correction flag for p, i dir'n, (integer)
1               | iblkjp, block correction flag for p, j dir'n, (integer)
1.8             | rlx, relaxation parameter used in tdma2d, (real)
50.             | et, E factor for the temperature equation, (real)
50.             | eu, E factor for the u momentum equation, (real)
50.             | ev, E factor for the v momentum equation, (real)
1.8             | rlxt, relaxation parameter for temperature, (real)
300             | ntry, number of loops through momentum solution
3               | jplotb, vertical position to begin plotting, (integer)
2               | jplote, vertical position to end plotting, (integer)
00              | iopat, node/element output flag for PATRAN neutral file
41              | jpate, PATRAN files generated for j = jb to jpate (integer)
3 3 1           | iflgcds, do loop parameters, 1=uds, 2=transition, 3=cds
1.0             | ecds, E factor for t,u,v after cds introduced (real)
0  0  0         | ipert, iteration number of perturbation for each of 3 loops
1 1             | irwtuv, read/write flags for tuv data in/out at start/end
0               | npert, number of cells to be perturbed (0 for automatic)
1               | iwtofy, =1 to write vertical temp. dist'n to file (integer)
10              | irepfrq, report frequency for output file (integer)
1               | ioqflg,=1 to write local heat flux data to outut file (int)
```

```
0.00            ! rwws, specular component of LW reflectance, west wall
0.00            ! rews, specular component of LW reflectance, east wall
0.16            ! rwwd, diffuse component of LW reflectance, west wall
0.953           ! rewd, diffuse component of LW reflectance, east wall
0.946           ! rseal, LW reflectance of seals exposed to cavity
0.0635          ! wideg, defined width of edge-glass from sight line [m]
1               ! idorad, flag to switch radiation off if zero (integer)
```

Appendix C

Sample IGU-2D Output File:
Results for Glazing System No. 3, Table 7.1

```
IGU3, Cardinal unit #1, tested Oct. 5, 1989, 3.0/14.0/3.0E2
klin = 2.0 W/m.C
May 25, 1989
Cavity Aspect Ratio, A=   43.40652
ib,ie=            2           30
ibc,iec=            4            28
nx ,nxc=           29            25
jb,je=            2          228
jbc,jec=            7          223
jbcg,jecg=            30          200
ny ,nyc=          227          217
kseal=   2.560000    [W/mK]
wseal=   11.05000    [mm]
nyseal=             5
dyseal=   2.210000    [mm]
kg=  0.9600000    [W/mK]
tg=   3.300000    [mm]
kn=  0.1700000    [W/mK]
tn=   1.524000    [mm]
rwws, rews  0.0000000E+00  0.0000000E+00
rwwd, rewd  0.1600000      0.9530000
rseal  0.9460000
jbcg, jecg          30          200
lcg            24
rcg            96
nsurf           144
west and east plate temps.=  0.8800000       20.00000      [C]
pane spacing=   14.12000      [mm]
gas no.=            2
Agrid=   5.000751              29   x         227  control volumes
diep  5.6479999E-04 [m]
djnp  2.8244241E-03 [m]
IFLGCDS parameters=           3           3           1

Resid Redn Sought        Max Itns     Block Correction Flags
T   10000.00              10           0            0
u   10000.00               4           0            0
v   10000.00               5           0            0
p   10000.00               7           0            1

Exchange factor check sums
          Max   1.001936
          Min   1.000000

 ITN  Log Resid.Ratio (b/a)  #     Average Mass    HEAT FLUX [W/sqm]
      T     u     v     pp   pp     Divergence       Avg (max-  Hot   do  no
                                  before    after          min%) Wall  cds sc
 All E-factors set to    1.000000    - start loop              3
   1  1.2*  5.9   5.3   4.4   14  0.7E-06  0.2E-10   55.6  0.1  55.6  1.0 16
  10  1.2*  3.8*  2.9*  4.7   14  0.7E-09  0.3E-11   55.6  0.1  55.6  1.0 16
  20  1.1*  3.7*  2.8*  4.7   14  0.6E-09  0.3E-11   55.6  0.1  55.6  1.0 16
  30  1.2*  3.7*  2.7*  4.2   14  0.5E-09  0.4E-11   55.6  0.2  55.6  1.0 16
  40  1.2*  3.7*  2.8*  4.3   14  0.5E-09  0.4E-11   55.6  0.2  55.6  1.0 16
  50  1.2*  3.6*  2.8*  4.5   14  0.5E-09  0.3E-11   55.7  0.3  55.6  1.0 16
 60  1.2*  3.6*  2.8*  4.6   14  0.4E-09  0.4E-11   55.7  0.3  55.6  1.0 16
 70  1.2*  3.5*  2.8*  4.3   14  0.3E-09  0.3E-11   55.7  0.3  55.6  1.0 16
 80  1.2*  3.5*  2.8*  4.3   14  0.3E-09  0.3E-11   55.7  0.3  55.6  1.0 16
 90  1.1*  3.4*  2.8*  4.2   14  0.2E-09  0.4E-11   55.7  0.3  55.6  1.0 16
100  1.1*  3.3*  2.7*  4.5   14  0.2E-09  0.4E-11   55.7  0.3  55.6  1.0 16
110  1.1*  3.3*  2.7*  4.5   14  0.2E-09  0.3E-11   55.7  0.3  55.6  1.0 16
120  1.1*  3.2*  2.7*  3.7*  49  0.1E-09  0.3E-11   55.7  0.3  55.6  1.0 16
130  1.0*  3.1*  2.6*  3.2*  49  0.1E-09  0.4E-11   55.7  0.3  55.6  1.0 16
140  1.0*  3.0*  2.6*  3.5*  49  0.9E-10  0.4E-11   55.7  0.2  55.6  1.0 16
150  1.0*  3.0*  2.6*  3.6*  49  0.8E-10  0.3E-11   55.7  0.2  55.6  1.0 16
160  1.0*  3.0*  2.5*  4.0   14  0.8E-10  0.3E-11   55.7  0.2  55.7  1.0 16
170  1.0*  3.0*  2.5*  3.2*  49  0.9E-10  0.3E-11   55.7  0.2  55.7  1.0 16
180  1.0*  3.0*  2.4*  3.5*  49  0.1E-09  0.3E-11   55.7  0.2  55.7  1.0 16
190  1.0*  3.0*  2.4*  4.5   14  0.1E-09  0.4E-11   55.7  0.1  55.7  1.0 16
200  1.0*  3.0*  2.3*  4.5   14  0.1E-09  0.4E-11   55.7  0.1  55.7  1.0 16
210  1.0*  3.1*  2.3*  4.5   14  0.1E-09  0.3E-11   55.7  0.1  55.7  1.0 16
220  1.0*  3.1*  2.2*  4.1   14  0.1E-09  0.3E-11   55.7  0.1  55.7  1.0 16
230  1.0*  3.1*  2.2*  4.1   14  0.1E-09  0.3E-11   55.7  0.1  55.7  1.0 16
240  1.0*  3.1*  2.2*  4.2   14  0.1E-09  0.3E-11   55.7  0.1  55.7  1.0 16
250  1.0*  3.1*  2.2*  4.2   14  0.1E-09  0.3E-11   55.7  0.1  55.7  1.0 16
260  1.0*  3.1*  2.2*  4.1   14  0.1E-09  0.3E-11   55.7  0.1  55.7  1.0 16
270  1.0*  3.1*  2.2*  4.2   14  0.1E-09  0.3E-11   55.7  0.1  55.7  1.0 16
280  1.0*  3.1*  2.2*  3.5*  49  0.1E-09  0.4E-11   55.7  0.1  55.7  1.0 16
290  0.9*  3.1*  2.2*  4.1   14  0.1E-09  0.3E-11   55.7  0.1  55.7  1.0 16
300  1.0*  3.1*  2.2*  3.5*  49  0.1E-09  0.3E-11   55.7  0.1  55.7  1.0 16
pexmax   0.4290076
peymax    12.67970
mid-height deltat    18.47293     [C deg]
mid-height tm    10.44039      [C]

con=    1.6859386E-02        [W/mK]
cp=     521.7495             [J/KgK]
visc=   2.1881802E-05        [Kg/msec]
beta=   3.5261007E-03        [/K]
den=    1.716222             [Kg/m3]
Ra=    7490.803
Pr=  0.6771787
Gr=   11061.78
Critical Gr and Ra =   8921.520        6041.463
Number of secondary cells expected=          16
Wave number =    2.815058

Heat Flux Summary  [W/sqm]
predicted qelsh (bottom) =    45.62294    [W/sqm]
predicted qelsh (middle) =    27.95866    [W/sqm]
predicted qelsh (top)    =    33.54441    [W/sqm]
qshew   94.09376         28.05180        28.07230        28.08030
55.37398
qmax (vertical surface at east cavity wall)   1064.947
qbmax    55.73301
qbmin    55.67670
qbbar     55.72007
qbhot     55.67731
```

```
Max/Min source terms and max/min irradiance values  [W/sqm]
West Wall    6.285055      -4.263164      333.0783      326.7174
Top  Wall   -1.172177      -2.697463      333.9839      329.7056
East Wall   -3.211695      -4.443497      330.9705      322.2385
Bottom Wall  -1.182551      -2.670982      331.6163      327.6032

Bottom sight line temperature =     16.20194    [C]

Centre-Glass Summary
qrad=     4.521101        [W/sqm]
qtot = qelsh(2) =    27.95866      [W/sqm]
qcc =     23.43756      [W/sqm]
Nusselt =    1.064908
delta-tsurf(cg)=    18.43289      [C]
qrad (1-d calculated) =     4.446151     [W/sqm]
C-value =     1.497908      [W/sqm]
```

Appendix D

IGU-2D
Fortran Source Code

```
c
c                 IGU2.FOR             John L. Wright
c
c       2-D Heat Transfer in a Integrated Glazing Unit
c
c                                      August, 1989
c
c                Mainline
c
      real t(42,302), p(42,302)
      real u(42,302), v(42,302), theta, speed
      real pp(42,302)
c
      real rt(42,302), qt(42,302)
      real ru(42,302), qu(42,302)
      real rv(42,302), qv(42,302)
c
      real ate(42,302), atw(42,302), atn(42,302),ats(42,302)
      real atp(42,302), btp(42,302)
      real aue(42,302), auw(42,302), aun(42,302),aus(42,302)
      real aup(42,302), bup(42,302)
      real ave(42,302), avw(42,302), avn(42,302),avs(42,302)
      real avp(42,302), bvp(42,302)
      real ape(42,302), apw(42,302), apn(42,302),aps(42,302)
      real app(42,302), bpp(42,302)
c
      real det(42,302), dnt(42,302), con, diep, djnp, arep, arnp, cp
      real deu(42,302), dnu(42,302), dev(42,302), dnv(42,302)
c
      real deltax, deltay, lx,ly
      real kn, kg, kseal
c
      real rmep(42,302),rmnp(42,302)

      real rmeu(42,302), rmnu(42,302), rmev(42,302), rmnv(42,302)
c
      real alphae(42,302), alphan(42,302), betae(42,302), betan(42,302)
c
      real depp(42,302),dnpp(42,302)
      real mfrac(2)
      real qshew(5), qelsh(3)
c
      integer ib,ie,jb,je,id,jd, oput
      integer iflgcds(3), ipert(3), irwtuv(2)
c
      character*14 fnplot, fnoput
      character*8 srname
      character*1 ccon(2)
      character*80 title, subt(2)
c
c  radiation variables
      real amat(200,200), xvec(200), bvec(200)
      real alud(200,200)
      real sp(200)
      real hres(200)
      real tsurf(200)
      real e(200,200)
      integer tl, tr, br
      integer lcg, rcg
      integer indx(200)
      integer from, to
c
c
      common/IO/ iou,KYBD,IPLOT
c
      ccon(1)='*'
      ccon(2)=' '
c
      id=42
      jd=302
      matdim=200
c
c   Finite volume output results sent to unit 4
c   HP 7470 a pen plotter output data sent to unit 7
c   FEM node/element definition neutral file sent to unit 3
c   FEM nodal t/u/v data neutral file sent to unit 8
c   Console output sent to unit 6
c   Keyboard input read from unit 5
c   TUV arrays written for storage to unit 14
c   TUV arrays read from storage from unit 15
c   Stream function results (x,y,phi) written in ASCII to unit 17
c   Vertical temperature distribution written to file at unit 18
c   Local heat flux distribution written to file at unit 19
c   Heat flux distribution for east plate written to file at unit 20
      oput=4
      iplot=7
      ipatnoel=3
      ipattuv=8
      iwty=18
      iou=6
      kybd=5
      iwq=19
      iwqep=20
c
      pi=3.1415927
c
      call input(ib,ie,jb,jbc,tww,twe,lx,ly,pb,
     &           idgas,mfrac,speed,theta,
     &           nitert,niteru,niterv,niterp,
     &           tolt,tolu,tolv,tolp,
     &           iblkit,iblkjt,iblkiu,iblkju,
     &           iblkiv,iblkjv,iblkip,iblkjp,
     &           rlx,et,eu,ev,rlxt,
     &           ntry,fnplot,fnoput,
     &           jplotb,jplote,iopat,jpate,title,subt,iflgcds,
     &           ecds,ipert,irwtuv,npert,iwtofy,irepfrq,
     &           kseal,wseal,kg,tg,kn,tn,ioqflg,
     &           rwws,rews,rwwd,rewd,rseal,wideg,idorad)
```

```
      aratio=ly/lx
            Grcrit=(1.+(5./Aratio))/1.25e-4
      ibc=ib+2
      iec=ie-2
      nx=ie-ib+1
      nxc=iec-ibc+1
      nyc=ifix(nxc*aratio/5.)
      agrid=aratio*float(nxc)/float(nyc)
      nyseal=jbc-jb
      jec=jbc+nyc-1
      je=jec+nyseal
      dyseal=wseal/float(nyseal)
      ny=je-jb+1
      ipref=(ibc+iec)/2
      jpref=(jbc+jec)/2
c
      call intial(atp,ate,atw,atn,ats,u,v,rmep,rmnp,det,dnt,
     &            p,t,speed,theta,tww,twe,
     &            ib,ie,jb,je,id,jd)
      if(irwtuv(1).eq.1) then
        read(15,*) t
        read(15,*) u
        read(15,*) v
      endif
c
      deltax=lx/float(iec-ibc+1)
      deltay=ly/float(jec-jbc+1)
c
      diep=deltax
      djnp=deltay
      arep=deltay
      arnp=deltax
c
      dieu=deltax
      djnu=deltay
      areu=deltay
      arnu=deltax
c
      diev=deltax
      djnv=deltay
      arev=deltay
      arnv=deltax
c
c  Calculate geometric exchange factors and A matrix for
c  long-wave radiation analysis
c
      jbcg=ifix((wideg/ly)*float(jec-jbc+1)) + 1 + jbc
      jecg=jec-(jbcg-jbc)
c
      call exfact(e,nsurf,tl,tr,br,lx,ly,rwws,rews,
     &                    lcg,rcg,
     &                    ibc,iec,jbc,jec,
     &                    jbcg,jecg,matdim,ierr,esumax,esumin,oput)
      if(ierr.eq.1) then
```

```
         write(oput,*) 'Matdim not large enough - terminated'
         goto 99
      endif
c
      call setupa(amat,e,rwwd,rewd,rseal,
     &            matdim,tl,tr,br,nsurf)
c
      do 82 i=1,matdim
        do 83 j=1,matdim
          alud(i,j)=amat(i,j)
83      continue
82    continue
      call ludcmp(alud,nsurf,matdim,indx,radumy)
c
c
c     open(unit=oput, file=fnoput, status='old')
c
      write(oput,*) title
      write(oput,*) subt(1)
      write(oput,*) subt(2)
      write(oput,*) 'Cavity Aspect Ratio, A=',aratio
      write(oput,*) 'ib,ie=',ib,ie
      write(oput,*) 'ibc,iec=',ibc,iec
      write(oput,*) 'nx ,nxc=',nx ,nxc
      write(oput,*) 'jb,je=',jb,je
      write(oput,*) 'jbc,jec=',jbc,jec
      write(oput,*) 'jbcg,jecg=',jbcg,jecg
      write(oput,*) 'ny ,nyc=',ny ,nyc
      write(oput,*) 'kseal=',kseal,'[W/mK]'
      write(oput,*) 'wseal=',wseal *1000.,'[mm]'
      write(oput,*) 'nyseal=',nyseal
      write(oput,*) 'dyseal=',dyseal*1000.,'[mm]'
      write(oput,*) 'kg=',kg,'[W/mK]'
      write(oput,*) 'tg=',tg *1000.,'[mm]'
      write(oput,*) 'kn=',kn,'[W/mK]'
      write(oput,*) 'tn=',tn *1000.,'[mm]'
      write(oput,*) 'rwws, rews', rwws, rews
      write(oput,*) 'rwwd, rewd', rwwd, rewd
      write(oput,*) 'rseal', rseal
      write(oput,*) 'jbcg, jecg', jbcg, jecg
      write(oput,*) 'lcg', lcg
      write(oput,*) 'rcg', rcg
      write(oput,*) 'nsurf', nsurf
      write(oput,*) 'west and east plate temps.=',tww, twe, '  [C]'
      write(oput,*) 'pane spacing=',lx*1000., '  [mm]'
      write(oput,*) 'gas no.=',idgas
      write(oput,*) 'Agrid=',Agrid,nx,'   x',ny,'  control volumes'
      write(oput,*) 'diep', diep, ' [m]'
      write(oput,*) 'djnp', djnp, ' [m]'
c
      write(oput,*) 'IFLGCDS parameters=',iflgcds(1),iflgcds(2),
     &                                    iflgcds(3)
      write(oput,*)
      write(oput,*) 'Resid Redn Sought       Max Itns
     &               ' Block Correction Flags'
      write(oput,*) 'T',tolt,nitert,iblkit,iblkjt
      write(oput,*) 'u',tolu,niteru,iblkiu,iblkju
      write(oput,*) 'v',tolv,niterv,iblkiv,iblkjv
      write(oput,*) 'p',tolp,niterp,iblkip,iblkjp
      write(oput,*)
      write(oput,*) 'Exchange factor check sums'
      write(oput,*) '         Max',esumax
      write(oput,*) '         Min',esumin
      write(oput,*)
61    format(1x,i3,1x,4(f4.1,a1,1x),i3,1x,2( e8.1,1x),1x,f5.1,
     &        1x,f4.1,1x,f5.1,2x,f3.1,1x,i2)
      write(oput,62)
      write(oput,63)
      write(oput,64)
62    format(2x,'ITN  Log Resid.Ratio (b/a)  #     Average Mass ',
     &          '   HEAT FLUX [W/sqm]')
63    format(2x,'      T      u     v     pp   pp     Divergence   ',
     &           '     Avg (max-  Hot  do  no.')
64    format(2x,'                                  before     after',
     &           '         min%) Wall  cds ss')
c
      do 101 itry=iflgcds(1),iflgcds(2),iflgcds(3)
c                     <------------------------------------
c                                                          |
c                                                          |
c  itry=1 loop does ntry iterations using uds scheme
c  itry=2 loop does ntry iterations while gradually introducing cds
c  itry=3 loop does ntry iterations to grind in cds solution
c
      if(itry.gt.1) then
      et=ecds
      ev=ecds
      eu=ecds
      write(oput,*) ' All E-factors set to ',ecds,
     &              ' - start loop',itry
c
      endif
c
      docds=0.
      do 100 iter=1,ntry
c                     <------------------------------------
c                                                          |
c                                                          |
      call tcav(t,tm,deltat,
     &          con,diep,kg,tg,
     &          ibc,iec,jbc,jec)
      call fra(Ra,Tm+273.159,Pb,1x ,deltat,idgas,mfrac,
     &         con,cp,visc,beta,den)
                     Pr=  visc*cp/con
                     Racrit=pr*grcrit
                     Gr=Ra/Pr
c
      docds=float(iter)/float(ntry)
      if(itry.eq.1) docds=0.
      if(itry.eq.3) docds=1.0
c
      icont=0
      iconu=0
      iconv=0
      iconp=0
c  Calculate mass flow rates at all faces in cavity using
c  currently held velocity components (u,v) and density of
c  the fill gas (den).  The subroutine massp has been modified
c  to manually set the values of rmep and rmnp equal to zero
c  along and outside the walls of the cavity.
c
      call massp(rmep,rmnp,u,v,den,arep,arnp,
     &           ibc,iec,jbc,jec,id,jd,oput)
c
c  Calculate the diffusion terms, first at the interior faces
c  of the control volumes in the fill gas cavity and then at
c  all remaining faces in or bordering on the solid material.
      call dt(det,dnt,con,diep,djnp,arep,arnp,cp,
     &           ibc,iec,jbc,jec,id,jd)
      call dtsol(det,dnt,con,diep,djnp,arep,arnp,cp,
     &           dyseal,kseal,kg,tg,kn,tn,
     &           ibc,iec,jbc,jec,
     &           ib,ie,jb,je,id,jd)
c
c  Calculate the upstream weighting factors for all control
c  volume faces - including those in the solid material - based
c  on the current mass flows and diffusion terms at each face.
      call weight(alphae,alphan,betae,betan,pexmax,peymax,
     &           rmep,rmnp,det,dnt, docds,
     &           ib,ie,jb,je,id,jd)
c
c  Calculate source terms at cavity walls to account for long-
c  wave radiation exchange
c
      call setemp(tsurf,t,nsurf,con,kg,kseal,
     &            diep,djnp,tg,dyseal,
     &            ibc,iec,jbc,jec,jbcg,jecg,oput)
c
      call setrhs(bvec,e,tsurf,rews,rwws,rewd,rwwd,rseal,
     &           tl,tr,br,nsurf)
c
      do 72 i=1,matdim
        xvec(i)=bvec(i)
72    continue
      call lubksb(alud,nsurf,matdim,indx,xvec)
c
c     call mprove(amat,alud,nsurf,matdim,indx,bvec,xvec)
      call calcres(hres,amat,xvec,bvec,nsurf)
c
      call calcsp(sp,xvec,tsurf,rews,rwws,rewd,rwwd,rseal,
     &                  tl,tr,br,nsurf)
c
```

```
      call checksp(spsum,sp,lx,ly,
     &             ibc,iec,jbc,jec,jbcg,jecg,
     &             tl,tr,br,lcg,rcg,nsurf)
c
      if(abs(spsum).gt.0.1)
     &      write(oput,*) iter,' wall source checksum',spsum,
     &                         ' [watts]'
c
      call sorct(rt,qt,ib,ie,jb,je,id,jd)
      if(idorad.ne.0)
     &call convsp(sp,rt,qt,tg,dyseal,cp,diep,djnp,
     &            ib,ie,jb,je,
     &            ibc,iec,jbc,jec,jbcg,jecg,
     &            tl,tr,br,lcg,rcg,nsurf,oput)
c
      call coeff(atp,ate,atw,atn,ats,btp,rt,qt,t,
     &           det,dnt,rmep,rmnp,alphae,alphan,betae,betan,et,
     &           ib,ie,jb,je,id,jd)
c
      call bndct(atp,ate,atw,atn,ats,btp,
     &           alphae,alphan,betae,betan,tww,twe,
     &           ib,ie,jb,je,id,jd)
c
      call bcelim(atp,ate,atw,atn,ats,btp,
     &           1,0.,t,
     &           ib,ie,jb,je,id,jd)
c
      call phislv(t,atp,ate,atw,atn,ats,btp,rlxt,nitert,icont,
     &            tolt,idone,reso,res,alrt,iblkit,iblkjt,
     &           ib,ie,jb,je,id,jd)
c  Drive in the temperature field one extra time on the first
c  iteration following perturbation of the velocity field.
           if((iter.eq.ipert(itry)+1).and.(iter.ne.1))then
           call phislv(t,atp,ate,atw,atn,ats,btp,rlxt,nitert,icont,
     &                 tolt,idone,reso,res,alrt,iblkit,iblkjt,
     &                ib,ie,jb,je,id,jd)
           endif
c
      call null(bup,ib,ie,jb,je,id,jd)
      call massu(rmeu,rmnu,rmep,rmnp,
     &           ibc,iec,jbc,jec,id,jd)
      call du(deu,dnu,visc,dieu,djnu,areu,arnu,
     &        ibc,iec,jbc,jec,id,jd)
      call weight(alphae,alphan,betae,betan,dvx,dvy,
     &            rmeu,rmnu,deu,dnu, docds,
     &            ibc,iec-1,jbc,jec,id,jd)
      call sorcu(ru,qu,
     &           ibc,iec,jbc,jec,id,jd)
      call coeff(aup,aue,auw,aun,aus,bup,ru,qu,u,
     &           deu,dnu,rmeu,rmnu,alphae,alphan,betae,betan,eu,
     &           ibc,iec-1,jbc,jec,id,jd)
      call sorcpu(bup,p,areu,
     &            ibc,iec,jbc,jec,id,jd)
c
```

```
      call bndcu(aup,aue,auw,aun,aus,bup,
     &             alphae,alphan,betae,betan,
     &             ibc,iec,jbc,jec,id,jd)
c
      call bcelim(aup,aue,auw,aun,aus,bup,
     &             1,0.,u,
     &             ibc,iec-1,jbc,jec,id,jd)
c
c
      call null(bvp,ib,ie,jb,je,id,jd)
      call massv(rmev,rmnv,rmep,rmnp,
     &             ibc,iec,jbc,jec,id,jd)
      call dv(dev,dnv,visc,diev,djnv,arev,arnv,
     &          ibc,iec,jbc,jec,id,jd)
      call weight(alphae,alphan,betae,betan,dvx,dvy,
     &              rmev,rmnv,dev,dnv, docds,
     &              ibc,iec,jbc,jec-1,id,jd)
      call sorcv(rv,qv,
     &             ibc,iec,jbc,jec,id,jd)
      call coeff(avp,ave,avw,avn,avs,bvp,rv,qv,v,
     &             dev,dnv,rmev,rmnv,alphae,alphan,betae,betan,ev,
     &             ibc,iec,jbc,jec-1,id,jd)
      call sorcpv(bvp,p,arnv,
     &              ibc,iec,jbc,jec,id,jd)
      call srcvnc(bvp,t,den,beta,arev,arnv,twe,tww,
     &              ibc,iec,jbc,jec,id,jd)
c
      call bndcv(avp,ave,avw,avn,avs,bvp,
     &             alphae,alphan,betae,betan,
     &             ibc,iec,jbc,jec,id,jd)
c
      call bcelim(avp,ave,avw,avn,avs,bvp,
     &             1,0.,v,
     &             ibc,iec,jbc,jec-1,id,jd)
c
          rlxu=rlxt
      call phislv(u,aup,aue,auw,aun,aus,bup,rlxu,niteru,iconu,
     &              tolu,idone,reso,res,alru,iblkiu,iblkju,
     &             ibc,iec-1,jbc,jec,id,jd)
c
          rlxv=rlxt
      call phislv(v,avp,ave,avw,avn,avs,bvp,rlxv,niterv,iconv,
     &             tolv,idone,reso,res,alrv,iblkiv,iblkjv,
     &             ibc,iec,jbc,jec-1,id,jd)
c
      if(ntry.eq.12345) goto 97
c   suppress pp corrections if ntry=12345 has been specified
      call coefp(app,ape,apw,apn,aps,
     &             aup,aue,auw,aun,aus,
     &             avp,ave,avw,avn,avs,
     &             arep,arnp,den,
     &             depp, dnpp,
     &             ibc,iec,jbc,jec,id,jd)
c
```

```
      call massp(rmep,rmnp,u,v,den,arep,arnp,
     &           ibc,iec,jbc,jec,id,jd,oput)
      call sorcp(bpp,rmep,rmnp,
     &           ibc,iec,jbc,jec,id,jd)
      call pest(pp,ib,ie,jb,je,id,jd)
c
      call massch (bpp,bpavgb,bpmaxb,bptotb,
     &           ibc,iec,jbc,jec,id,jd)
c
        call phislv(pp,app,ape,apw,apn,aps,bpp,rlxt,niterp,iconp,
     &                tolp,idonep,reso,res,alrp,iblkip,iblkjp,
     &                ibc,iec,jbc,jec,id,jd)
c
      call pshift(pp,ipref,jpref,ibc,iec,jbc,jec,id,jd)
      call velcor(u,v,depp,dnpp,pp,ibc,iec,jbc,jec,id,jd)
c
c  Calculate u at fictitious boundary condition volumes
c
c       call phislv(u,aup,aue,auw,aun,aus,bup,rlxu,0,idv,
c    &              tolu,idone,reso,res,alru,iblkiu,iblkju,
c    &              ib,ie-1,jb,je,id,jd)
c
c  Calculate v at fictitious boundary condition volumes
c
c       call phislv(v,avp,ave,avw,avn,avs,bvp,rlxv,0,idv,
c    &              tolv,idone,reso,res,alrv,iblkiv,iblkjv,
c    &              ib,ie,jb,je-1,id,jd)
c
      call pcor(p,pp,ibc,iec,jbc,jec,id,jd)
c
c
      call massp(rmep,rmnp,u,v,den,arep,arnp,
     &           ibc,iec,jbc,jec,id,jd,oput)
      call sorcp(bpp,rmep,rmnp,
     &           ibc,iec,jbc,jec,id,jd)
      call massch (bpp,bpavga,bpmaxa,bptota,
     &           ibc,iec,jbc,jec,id,jd)
c
      idbg=0
      call calcqb(qbbar,qbmax,qbmin,qbhot,rmep,det,t,u,tww,twe,
     &                   con,den,cp,
     &                   kseal,kg,kn,
     &                   diep,djnp,tg,tn,dyseal,ly,wseal,
     &                   sp,jbcg,jecg,lcg,rcg,tl,tr,br,nsurf,
     &                   ibc,iec,jbc,jec,
     &                   ib,ie,jb,je,id,jd, docds,oput,idbg)
      bandpc=100.*(qbmax-qbmin)/qbbar
c
      call cells(ncells,u,v,uclmax,ibc,iec,jbc,jec)
c
      itest = iter - (iter/irepfrq)*irepfrq
      if((iter.eq.1).or.(itest.eq.0).or.(iter.eq.ntry))
     &write(oput,61) iter,
     &                alrt,ccon(icont+1),alru,ccon(iconu+1),
     &               alrv,ccon(iconv+1),alrp,ccon(iconp+1),
     &               idonep,
     &               bpavgb,bpavga,
     &               qbbar,bandpc,qbhot,docds,ncells
c
c  Check value of ipert(itry) to see if perturbation has been
c  prescribed.
      idv=0
      if((itry.eq.1).and.(iter.eq.ipert(1))) idv=1
      if((itry.eq.2).and.(iter.eq.ipert(2))) idv=1
      if((itry.eq.3).and.(iter.eq.ipert(3))) idv=1
      if(idv.eq.1) then
                     nkorp=0
                     npcall = 0
         call iexpec(Gr,Aratio,nkorp,wavnum,pi)
         if((Aratio.le.12).or.(Gr.lt.Grcrit)) goto 666
         npcall=nkorp
         wncall=wavnum
         if(npert.ge.2) then
           npcall=npert
           wncall=wavnum*float(npert)/float(nkorp)
         endif
      call perturb(npcall,wncall,uclmax,aratio,u,v,pi,ibc,iec,jbc,jec)
666   write(oput,*) npcall,' secondary cells perturbed'
      write(oput,*) ' Wave number of perturbed cells =',wncall
      endif
c
c                                                        ^
c                                                        |
c   ----------------------------------------------------
100   continue
c
97    write(oput,*)'pexmax', pexmax
      write(oput,*)'peymax',peymax
c
      write(oput,*) 'mid-height deltat',deltat,' [C deg]'
      write(oput,*) 'mid-height tm',tm,' [C]'
c
      write(oput,*)
      write(oput,*) 'con= ',con,'        [W/mK]'
      write(oput,*) 'cp= ',cp,'         [J/KgK]'
      write(oput,*) 'visc=',visc,'         [Kg/msec]'
      write(oput,*) 'beta=',beta,'         [/K]'
      write(oput,*) 'den=',den,'          [Kg/m3]'
      write(oput,*) 'Ra=',ra
                     Pr=  visc*cp/con
      write(oput,*) 'Pr=',Pr
      write(oput,*) 'Gr=',Gr
      write(oput,*) 'Critical Gr and Ra =',Grcrit, Racrit
                     nkorp=0
                     if(Aratio.le.12) goto 66
                     if(Gr.lt.Grcrit) goto 66
                     call iexpec(Gr,Aratio,nkorp,wavnum,pi)
66    write(oput,*) 'Number of secondary cells expected=', nkorp
```

```
      if(nkorp.gt.1) write(oput,*) 'Wave number = ',wavnum
      write(oput,*)
      call fqep(t,twe,tww,iwqep,qshew,qelsh,qmaxep,
     &          kn,tn,jbc,jec,ly,wseal,jbcg,jecg,
     &          ib,ie,jb,je,id,jd)
c     write(oput,*)'yelshs',yelshs
c     write(oput,*)'yelshn',yelshn
c     write(oput,*)'yshews',yshews
c     write(oput,*)'yshewn',yshewn
      write(oput,*)'Heat Flux Summary  [W/sqm]'
c     write(oput,*)'qelsh',qelsh
      write(oput,*)'predicted qelsh (bottom) = ', qelsh(1) , '[W/sqm]'
      write(oput,*)'predicted qelsh (middle) = ', qelsh(2) , '[W/sqm]'
      write(oput,*)'predicted qelsh (top)    = ', qelsh(3) , '[W/sqm]'
      write(oput,*)'qshew',qshew
c
101   continue
c
c                                                        ^
c   -----------------------------------------------------|
c
98    call      fqofy(t,twe,tww,qmax,ibc,iec,jbc,jec,
     &                ly,wseal,kg,kseal,con,kn,tg,diep,
     &                sp,tr,rcg,br,jbcg,jecg,
     &                ib,ie,jb,je,id,jd,ioqflg,iwq)
      write(oput,*)'qmax (vertical surface at east cavity wall)',qmax
      write(oput,*)'qbmax',qbmax
      write(oput,*)'qbmin',qbmin
      write(oput,*)'qbbar ',qbbar
      write(oput,*)'qbhot ',qbhot
c
      write(oput,*)
      write(oput,*) 'Max/Min source terms ',
     &              'and max/min irradiance values  [W/sqm]'
      call spscan(sp,1,tl-1,spmax,spmin)
      call spscan(xvec,1,tl-1,hirmax,hirmin)
       write(oput,*) 'West Wall', spmax,spmin,hirmax,hirmin
      call spscan(sp,tl,tr ,spmax,spmin)
      call spscan(xvec,tl,tr ,hirmax,hirmin)
       write(oput,*) 'Top  Wall', spmax,spmin,hirmax,hirmin
      call spscan(sp,tr+1,br-1,spmax,spmin)
      call spscan(xvec,tr+1,br-1,hirmax,hirmin)
       write(oput,*) 'East Wall', spmax,spmin,hirmax,hirmin
      call spscan(sp,br,nsurf ,spmax,spmin)
      call spscan(xvec,br,nsurf ,hirmax,hirmin)
       write(oput,*) 'Bottom Wall', spmax,spmin,hirmax,hirmin
      write(oput,*)
c
      write(oput,*) 'Bottom sight line temperature = ',t(iec+1,jbc),
     &              '[C]'
      write(oput,*)
c
      write(oput,*) 'Centre-Glass Summary'
      write(oput,*) 'qrad= ',sp(lcg),'  [W/sqm]'
      write(oput,*) 'qtot = qelsh(2) =',qelsh(2),' [W/sqm]'
      qcc = qelsh(2)-sp(lcg)
      write(oput,*) 'qcc = ',qcc,' [W/sqm]'
      cgnu = qcc*lx/(con*(tsurf(rcg)-tsurf(lcg)))
      write(oput,*) 'Nusselt =',cgnu
      write(oput,*) 'delta-tsurf(cg)=',tsurf(rcg)-tsurf(lcg),' [C]'
      epsl=1.-rwws-rwwd
      epsr=1.-rews-rewd
      qrld=fqrld(tsurf(lcg),tsurf(rcg),epsl,epsr)
      write(oput,*) 'qrad (1-d calculated) = ',qrld, '[W/sqm]'
      cgcval=fcgc(qelsh,deltat,tg,kg)
      write(oput,*) 'C-value = ',cgcval,' [W/sqm]'
c
      call ftofy(t,tmin,tmax,iwty,iwtofy,iec+1,
     &           ly,wseal,
     &           ibc,iec,jbc,jec,jbcg,jecg,
     &           ib,ie,jb,je,id,jd)
C     call CUV(U,V,deltax, deltay,
c    &         fnplot,
c    &         IB,IE,jplotb,jplote,ID,JD)
c
      call strfun(u,v,aratio,ibc,iec,jbc,jec)
c
      if(irwtuv(2).eq.1) then
        write(14,*) t
        write(14,*) u
        write(14,*) v
       endif
c
      if(iopat.eq.0) goto 99
      call nodel(lx,ly,ib,ie,jb,je,jpate,tww,twe,ipatnoel,
     &             wseal,nyseal,ibc,iec,jbc,jec,tn,tg,
     &             title)
      call notuv(t,u,v,ib,ie,jb,je,jpate,tww,twe,ipattuv,
     &             wseal,nyseal,ibc,iec,jbc,jec,tn,tg,
     &             kn,kg,con,lx,ly,
     &             title,subt)
99    continue
c     close(oput)
c     close(iou)
c     close(kybd)
      continue
      stop
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C  End of Mainline                                                  \
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C
      subroutine resid(resphi,phi,ap,ae,aw,an,as,b,
     &                 ib,ie,jb,je,id,jd)
C
      real phi(42,302)
      real ae(42,302), aw(42,302), an(42,302),as(42,302)
      real ap(42,302), b(42,302)
      sumsqu=0.
      do 10 i=ib,ie
        do 20 j=jb,je
          r = ap(i,j) * phi(i,j)
          r = r - ae(i,j)*phi(i+1,j) - aw(i,j)*phi(i-1,j)
          r = r - an(i,j)*phi(i,j+1) - as(i,j)*phi(i,j-1) - b(i,j)
          sumsqu = sumsqu + r**2
20      continue
10    continue
      resphi = sumsqu**0.5
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine massch (bp,bpavg,bpmax,bptot,
     &                  ib,ie,jb,je,id,jd)
c
      real bp(42,302)
c  bp is the net influx of mass at volume i,j
c  bptot is the total influx of [illegible]   for the problem domain and
c  is (hopefully) equal to zero
      bptot = 0.
      imax = 0
      jmax = 0
      bpmax = 0.
      bpavg = 0.
      do 10 i=ib,ie
        do 20 j=jb,je
          bptot=bptot+bp(i,j)
          absbp=abs(bp(i,j))
          absbpm=abs(bpmax)
          bpavg=bpavg+absbp
          if(absbp.le.absbpm) goto 20
          bpmax=bp(i,j)
          imax=i
          jmax=j
20      continue
10    continue
      bpavg=bpavg/(float(ie-ib+1)*float(je-jb+1))
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine sorct(rt,qt,
     &                  ib,ie,jb,je,id,jd)
c
      real rt(42,302), qt(42,302)
      do 10 i=ib,ie
        do 20 j=jb,je
          rt(i,j)=0.
          qt(i,j)=0.
20      continue
10    continue
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine srcvnc(bvp,t,den,beta,arev,arnv,twe,tww,
     &                 ib,ie,jb,je,id,jd)
c
      real bvp(42,302), t(42,302)
      g=9.8
      tref=(twe+tww)/2.
      do 10 i=ib,ie
        do 20 j=jb,je-1
          tijv=(t(i,j)+t(i,j+1))/2.
          bvp(i,j)=bvp(i,j)+den*g*beta*(tijv-tref)*arev*arnv
20      continue
10    continue
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine phislv(t,atp,ate,atw,atn,ats,btp,rlx,niter,icon,
     &                  tol,idone,reso,rest,alr,iblki,iblkj,
     &                  ib,ie,jb,je,id,jd)
c
      real ate(42,302), atw(42,302), atn(42,302),ats(42,302)
      real atp(42,302), btp(42,302)
      real t(42,302),rlx
      integer ib,ie,jb,je,id,jd
      integer niter
      resrat=1.
        call resid(reso,t,atp,ate,atw,atn,ats,btp,
     &                 ib,ie,jb,je,id,jd)
c
c             write(4,*) 'reso=',reso
c
      idone=0
      icon=1
c  Note: the phi values at the fictitious boundary volumes
c        can be updated by calling phislv(....) with niter=0
      if(niter.lt.1) goto 95
c
      do 5 iter=1,niter
c
        if(iblki.eq.1)
     &  call bcori(t,atp,ate,atw,atn,ats,btp,
     &               ib,ie,jb,je,id,jd)
        if(iblkj.eq.1)
     &  call bcorj(t,atp,ate,atw,atn,ats,btp,
     &               ib,ie,jb,je,id,jd)
c
            idoadi = niter
            call tdma2d(t,atp,ate,atw,atn,ats,btp,rlx,idoadi,
     &                  ib,ie,jb,je,id,jd)
            idone=idone+idoadi
c
```

```
        call resid(rest,t,atp,ate,atw,atn,ats,btp,
     &                  ib,ie,jb,je,id,jd)
c
c             write(4,*) 'rest=',rest
c
        if(rest.le.1.e-15) goto 94
        resrat=reso/rest
c
c             write(4,*) 'resrat=',resrat,'tol=',tol
c
        if(resrat.gt. tol) goto 94
c
5     continue
c
      icon=0
c
94    alr=alog10(resrat)
c
c             write(4,*) 'alr=',alr
c
c
c  Calculate t at fictitious boundary condition volumes if
c  coefficients are available.
c
c                   if(2.gt.1) goto 210
c
95    do 110 i=ib,ie
        if(abs(atp(i,jb-1)).lt.1.e-12) goto 111
        t(i,jb-1)=(atn(i,jb-1)*t(i,jb)+btp(i,jb-1))/atp(i,jb-1)
111     if(abs(atp(i,je+1)).lt.1.e-12) goto 110
        t(i,je+1)=(ats(i,je+1)*t(i,je)+btp(i,je+1))/atp(i,je+1)
110     continue
      do 210 j=jb,je
        if(abs(atp(ib-1,j)).lt.1.e-12) goto 211
        t(ib-1,j)=(ate(ib-1,j)*t(ib,j)+btp(ib-1,j))/atp(ib-1,j)
211     if(abs(atp(ie+1,j)).lt.1.e-12) goto 210
        t(ie+1,j)=(atw(ie+1,j)*t(ie,j)+btp(ie+1,j))/atp(ie+1,j)
210     continue
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C
      subroutine tdma2d(t,atp,ate,atw,atn,ats,btp,rlx,niter,
     &                  ib,ie,jb,je,id,jd)
C
      real ate(42,302), atw(42,302), atn(42,302),ats(42,302)
      real atp(42,302), btp(42,302)
      real t(42,302),rlx
      real alpha(302), beta(302)
74    format('+',20x,i4,3x,i4,3x,i4,3x,i4)
c
      do 5 iter=1,niter
c
C  Sweep in the +ve x direction
```

```
        do 10 i=ib,ie
          do 20 j=1,jd
            alpha(j)=0.
            beta(j)=0.
20        continue
C  Calculate alpha and beta coefficients up the column
C  Note that at j=jb, ats(i,j)=0
          do 30 j=jb,je
            apij=atp(i,j)-ate(i,j)*(rlx-1.)
            bpij=atw(i,j)*t(i-1,j)+ate(i,j)*(t(i+1,j)-(rlx-1.)*
     &           t(i,j))+btp(i,j)
            denom=apij-ats(i,j)*alpha(j-1)
            alpha(j)=atn(i,j)/denom
            beta(j)=(ats(i,j)*beta(j-1)+bpij)/denom
30        continue
C  Back substitute for new temperatures down the column
C  Note that t(i,je)=beta(je) because atn(i,je)=0.
          do 40 jdv=jb,je
            j=je-jdv+jb
            t(i,j)=alpha(j)*t(i,j+1) + beta(j)
40        continue
c      write(6,74) iter, niter, i, j
10      continue
C
C  Sweep in the +ve y direction
        do 50 j=jb,je
          do 60 i=1,id
            alpha(i)=0.
            beta(i)=0.
60        continue
C  Calculate alpha and beta coefficients across the row
C  Note that at i=ib, atw(i,j)=0
          do 70 i=ib,ie
            apij=atp(i,j)-atn(i,j)*(rlx-1.)
            bpij=ats(i,j)*t(i,j-1)+atn(i,j)*(t(i,j+1)-(rlx-1.)*
     &           t(i,j))+btp(i,j)
            denom=apij-atw(i,j)*alpha(i-1)
            alpha(i)=ate(i,j)/denom
            beta(i)=(atw(i,j)*beta(i-1)+bpij)/denom
70        continue
C  Back substitute for new temperatures back across the row
C  Note that t(ie,j)=beta(ie) because ate(ie,j)=0.
          do 80 idv=ib,ie
            i=ie-idv+ib
            t(i,j)=alpha(i)*t(i+1,j) + beta(i)
80        continue
50      continue
5     continue
c
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine prfl(alpha,beta,pe, docds)
```

```
c
c  docds=0 gives uds
c  docds=1 gives cds
           acds=0.
           bcds=1.
           pesq=pe**2
           alpha=pesq/(10.+2.*pesq)
           beta=(1.+0.005*pesq)/(1.+0.05*pesq)
      alpha=(docds*acds)+(1.-docds)*alpha
      beta =(docds*bcds)+(1.-docds)*beta
99    return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine weight(alphae,alphan,betae,betan,pemaxx,pemaxy,
     &                  rme,rmn,de,dn, docds,
     &                  ib,ie,jb,je,id,jd)
c
      real de(42,302), dn(42,302)
      real rme(42,302),rmn(42,302)
      real alphae(42,302), alphan(42,302), betae(42,302), betan(42,302)
c
      call null(alphae,ib,ie,jb,je,id,jd)
      call null(betae,ib,ie,jb,je,id,jd)
      pemaxx=0.
      do 10 j=jb,je
        do 20 i=ib-1,ie
c
      if(abs(de(i,j)).lt.1.e-20) write(4,*)'east',i,j
c
          peij=rme(i,j)/de(i,j)
          if(peij.gt.pemaxx) pemaxx=peij
          call prfl(alphae(i,j),betae(i,j),peij, docds)
20      continue
10    continue
c
      call null(alphan,ib,ie,jb,je,id,jd)
      call null(betan,ib,ie,jb,je,id,jd)
      pemaxy=0.
      do 30 j=jb-1,je
        do 40 i=ib,ie
c
      if(abs(dn(i,j)).lt.1.e-20) write(4,*)'north',i,j
c
          peij=rmn(i,j)/dn(i,j)
          if(peij.gt.pemaxy) pemaxy=peij
          call prfl(alphan(i,j),betan(i,j),peij, docds)
40      continue
30    continue
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine input(ib,ie,jb,jbc,tww,twe,lx,ly,pb,
     &                 idgas,mfrac,speed,theta,
     &                 nitert,niteru,niterv,niterp,
     &                 tolt,tolu,tolv,tolp,
     &                 iblkit,iblkjt,iblkiu,iblkju,
     &                 iblkiv,iblkjv,iblkip,iblkjp,
     &                 rlx,et,eu,ev,rlxt,
     &                 ntry,fnplot,fnoput,
     &                 jplotb,jplote,iopat,jpate,title,subt,iflgcds,
     &                 ecds,ipert,irwtuv,npert,iwtofy,irepfrq,
     &                 kseal,wseal,kg,tg,kn,tn,ioqflg,
     &                 rwws,rews,rwwd,rewd,rseal,wideg,idorad)
c
      real lx,ly, mfrac(2)
      real kn, kg, kseal
      character*20 datfn
      character*14 fnplot ,fnoput
      character*80 title, subt(2)
      integer iflgcds(3), ipert(3), irwtuv(2)
25    format(a80)
      indat=9
      datfn='run.dat'
c     open(unit=indat,file=datfn,status='old')
      read(indat,25) title
      read(indat,25) subt(1)
      read(indat,25) subt(2)
      read(indat,*) ib
      read(indat,*) ie
      read(indat,*) jb
      read(indat,*) jbc
      read(indat,*) tww
      read(indat,*) twe
      read(indat,*) lx
      read(indat,*) ly
      read(indat,*) pb
      read(indat,*) idgas
      read(indat,*) mfrac(1)
                    mfrac(2)=1.- mfrac(1)
                    speed = 0.
                    theta = 0.
      read(indat,*) kseal
      read(indat,*) wseal
      read(indat,*) kg
      read(indat,*) tg
      read(indat,*) kn
      read(indat,*) tn
      read(indat,*) nitert
      read(indat,*) niteru
      read(indat,*) niterv
      read(indat,*) niterp
      read(indat,*) tolt
      read(indat,*) tolu
      read(indat,*) tolv
      read(indat,*) tolp
      read(indat,*) iblkit
      read(indat,*) iblkjt
      read(indat,*) iblkiu
      read(indat,*) iblkju
      read(indat,*) iblkiv
      read(indat,*) iblkjv
      read(indat,*) iblkip
      read(indat,*) iblkjp
      read(indat,*) rlx
      read(indat,*) et
      read(indat,*) eu
      read(indat,*) ev
      read(indat,*) rlxt
c     read(indat,*) ipref
c     read(indat,*) jpref
      read(indat,*) ntry
                    fnplot = 'vslot.plt'
                    fnoput = 'vslot.out'
      read(indat,*) jplotb
      read(indat,*) jplote
      read(indat,*) iopat
      read(indat,*) jpate
      read(indat,*) iflgcds(1), iflgcds(2), iflgcds(3)
      read(indat,*) ecds
      read(indat,*) ipert
      read(indat,*) irwtuv
      read(indat,*) npert
      read(indat,*) iwtofy
      read(indat,*) irepfrq
      read(indat,*) ioqflg
      read(indat,*) rwws
      read(indat,*) rews
      read(indat,*) rwwd
      read(indat,*) rewd
      read(indat,*) rseal
      read(indat,*) wideg
      read(indat,*) idorad
5     format(a14)
c     close(indat)
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine intial(atp,ate,atw,atn,ats,u,v,rmep,rmnp,det,dnt,
     &                   p,t,speed,theta,tww,twe,
     &                   ib,ie,jb,je,id,jd)
c
      real ate(42,302), atw(42,302), atn(42,302),ats(42,302)
      real atp(42,302)
      real t(42,302), p(42,302)
      real det(42,302), dnt(42,302)
      real rmep(42,302),rmnp(42,302)
      real u(42,302), v(42,302), theta, speed
      integer ib,ie,jb,je,id,jd
c
```

```
      call null(atp,ib,ie,jb,je,id,jd)
      call null(ate,ib,ie,jb,je,id,jd)
      call null(atw,ib,ie,jb,je,id,jd)
      call null(atn,ib,ie,jb,je,id,jd)
      call null(ats,ib,ie,jb,je,id,jd)
c
      call null(det,ib,ie,jb,je,id,jd)
      call null(dnt,ib,ie,jb,je,id,jd)
c
      call null(t,ib,ie,jb,je,id,jd)
      call null(p,ib,ie,jb,je,id,jd)
c
      call null(u,ib,ie,jb,je,id,jd)
      call null(v,ib,ie,jb,je,id,jd)
c
      call null(rmep,ib,ie,jb,je,id,jd)
      call null(rmnp,ib,ie,jb,je,id,jd)
c
      do 11 i=1,id
        tinit=tww+(twe-tww)*(float(i-ib)+0.5)/(float(ie-ib)+1.0)
        do 21 j=1,jd
          t(i,j)=tinit
21      continue
11    continue
c
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine pest(pp,ib,ie,jb,je,id,jd)
c
      real pp(42,302)
      integer ib,ie,jb,je,id,jd
      do 11 i=1,id
        do 21 j=1,jd
          pp(i,j)=0.
21      continue
11    continue
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine pinit(pp,ib,ie,jb,je,id,jd)
c
      real pp(42,302)
      integer ib,ie,jb,je,id,jd
      do 11 i=1,id
        do 21 j=1,jd
          pp(i,j)=0.
21      continue
11    continue
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine tdma(phi,ap,an,as,bp,
     &                jb,je,jd)
c
      real phi(302), ap(302), bp(302)
      real an(302), as(302)
      real alpha(302), beta(302)
      do 6 j=1,jd
        alpha(j)=0.
        beta(j)=0.
6     continue
      do 30 j=jb,je
        denom=ap(j) - as(j)*alpha(j-1)
        alpha(j)=an(j)/denom
        beta(j)=(as(j)*beta(j-1) + bp(j))/denom
30    continue
      do 40 jdv=jb,je
        j=je-jdv+jb
        phi(j)=alpha(j)*phi(j+1)+beta(j)
40    continue
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
c  This subroutine counts the number of secondary cells by logging
c  the number of times the u velocity component passes through zero
c  going positive to negative along the vertical centre line of the
c  vertical slot.
c  The maximum x-direction velocity component, u, along the vertical
c  centre-line is also logged and returned as umax in m/sec
      subroutine cells(ncells,u,v,umax,ib,ie,jb,je)
      real u(42,302), v(42,302)
      ncol=ie-ib+1
      rib=float(ib)
      rie=float(ie)
      icent=int(0.1 + (rib+rie)/2.)
      itest=2*(ncol/2)
      ileft=icent
      iright=icent
      if(itest.ne.ncol) ileft=icent-1
c
      umax=0.
      umin=0.
      do 20 j=jb,je
        uij=(u(ileft,j)+u(iright,j))/2.
        if(uij.gt.umax) umax=uij
        if(uij.lt.umin) umin=uij
20    continue
c
c       write(4,*) ncol,itest,icent,ileft,iright
      ncells=0
      uold=1.
      do 10 j=jb,je
        uij=(u(ileft,j) [illegible]right,j))/2.
```

```
          if((uij.gt.0.1*umin).and.(uij.lt.0.1*umax)) goto 10
          if((uij.lt.0.).and.(uold.ge.0.)) ncells=ncells+1
          uold=uij
10      continue
        if(ncells.eq.1) ncells=0
        return
        end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
c  This subroutine estimates the number of secondary cells expected
c  in the vertical slot as a function of aspect ratio, A, and Grashof
c  number, Gr, according to the data found in Lee & Korpela, 1983.
c  The corresponding wavenumber, alpha, is also returned.
c
        subroutine iexpec(Gr,A,nkorp,alpha,pi)
        alpha=2.82
        if(gr.gt.11000.) alpha=2.82+(2.50-2.82)*(gr-11000.)
       &                       /(15000.-11000.)
        if(gr.gt.15000.) alpha=2.50+(2.41-2.50)*(gr-15000.)
       &                       /(20000.-15000.)
        if(gr.gt.20000.) alpha=2.41+(2.33-2.41)*(gr-20000.)
       &                       /(25000.-20000.)
        if(gr.gt.25000.) alpha=2.33
        nkorp=0
        if(A.lt.12) goto 99
        nkorp=int((A-10.)*(alpha/(2.*pi))) + 2
c       if(nkorp.eq.1) nkorp=0
99      return
        end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
c  This subroutine superimposes nc secondary cells of wave number
c  alpha on top of the given base flow defined by u and v.
c  The maximum x-direction velocity added is umax.
c  The velocity field for the secondary cells is given by
c  the stream function PHI(x,y) = X(x)Y(y).  X is calculated
c  by fxfun and Y is calculated by fyfun.
c
        subroutine perturb(nc,alpha,umax,a,u,v,pi,ib,ie,jb,je)
        real u(42,302), v(42,302)
        if(nc.lt.2) goto 99
        ndivx=ie-ib+1
        ndivy=je-jb+1
        noe=nc-ifix((0.1+(float(nc)/2.)))*2
        hh=(float(nc)*pi)/(a*alpha)
         toplim=0.5+hh
         botlim=0.5-hh
        do 10 i=ib,ie
          xeovlx=float(i-ib+1)/float(ndivx  )
          xwovlx=float(i-ib  )/float(ndivx  )
          xfune=fxfun(xeovlx,pi)
          xfunw=fxfun(xwov[illegible]pi)
          do 20 j=jb,je
            ynovly=float(j-jb+1)/float(ndivy  )
        ysovly=float(j-jb  )/float(ndivy  )
        yfunn=fyfun(a,ynovly,alpha,noe,toplim,botlim,pi)
        yfuns=fyfun(a,ysovly,alpha,noe,toplim,botlim,pi)

        u(i,j)=u(i,j)+2.*xfune*(yfuns-yfunn)*ndivy*umax/(alpha*a)
        v(i,j)=v(i,j)+2.*yfunn*(xfune-xfunw)*ndivx*umax/alpha

20       continue
10    continue
99    return
      end

      function fxfun(xovlx,pi)
        fxfun= 0.5*(cos(2.*pi*(xovlx-0.5))+1.)
      return
      end

      function fyfun(a,yovly,alpha,noe,toplim,botlim,pi)
       fyfun=0.
        if(yovly.ge.toplim) goto 99
        if(yovly.le.botlim) goto 99
       arg=a*(yovly-0.5)*alpha + float(noe)*pi
       fyfun=-0.5*(1.-cos(arg))
99    return
      end
c
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
      subroutine strfun(u,v,aview,ib,ie,jb,je)
      real  phine(42,302)
      real u(42,302), v(42,302)
      character*4 ftype
77    format(8(1x,f9.6))
71    format(2(2x,i3))
72    format(a4)
      ftype='DSAA'
      zero=0.
      one=1.
      ndivx=ie-ib+1
      ndivy=je-jb+1
      deltay=1.
c
      phimin=1000.
      phimax=-1000.
c
      write(17,72) ftype
      write(17,71) ndivx+1, ndivy+1
      write(17,77) zero, one
      write(17,77) zero, aview
c
      j=jb-1
      do 30 i=ib-1,ie
       phine(i,j)=0.
       if(phine(i,j).gt.phimax) phimax=phine(i,j)
```

```
        if(phine(i,j).lt.phimin) phimin=phine(i,j)
30    continue
c
      do 10 j=jb,je-1
        i=ib-1
        phine(i,j)=0.
        if(phine(i,j).gt.phimax) phimax=phine(i,j)
        if(phine(i,j).lt.phimin) phimin=phine(i,j)
        do 20 i=ib,ie-1
            phine(i,j)=phine(i,j-1)-u(i,j)*deltay
            if(phine(i,j).gt.phimax) phimax=phine(i,j)
            if(phine(i,j).lt.phimin) phimin=phine(i,j)
20      continue
          i=ie
          phine(i,j)=0.
10    continue
      j=je
      do 40 i=ib-1,ie
        phine(i,j)=0.
40    continue
c
      write(17,77) phimin, phimax
c
      do 110 j=jb-1,je
            write(17,77) (phine(i,j),i=ib-1,ie)
110   continue
c
c
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
c  This subroutine records the heat flux [W/sqm] distribution as a
c  function of y (the vertical dimension) at the west face of the
c  right hand sheet of glass.
c
      subroutine fqofy(t,twe,tww,qmax,ibc,iec,jbc,jec,
     &                  ly,wseal,kg,kseal,con,kn,tg,diep,
     &                  sp,tr,rcg,br,jbcg,jecg,
     &                  ib,ie,jb,je,id,jd,ioflg,iwq)
c
      real ly,kg,kseal,keff,kn
      real t(42,302)
      real sp(200)
      integer tr,rcg,br,jbcg,jecg
66    format(2x,f8.4,2x,f8.2)
      zero=0.
      toth=ly + 2.*wseal
      qmax=0.
      dx1=diep/2.
      dx2=tg/2.
c  Adjacent bottom seal
      keff=fkeff(kseal,kg,dx1,dx2)
      do 10 j=jb,jbc-1
```

```
        ys=(float(j-jb)/float(jbc-jb))*wseal
        yn=(float(j-jb+1)/float(jbc-jb))*wseal
         ys=ys/toth
         yn=yn/toth
        q = keff * (t(iec+1,j) - t(iec,j))/(dx1+dx2)
        if(q.gt.qmax) qmax=q
        if(ioflg.ne.1) goto 10
        write(iwq,66) ys,q
        write(iwq,66) yn,q
10    continue
c  Adjacent cavity
      keff=fkeff(con,kg,dx1,dx2)
      do 20 j=jbc,jec
        isurf=rcg
        if(j.gt.jecg) isurf=rcg - (j-jecg)
        if(j.lt.jbcg) isurf=rcg + (jbcg-j)
        ys=wseal + ly * float(j-jbc)/float(jec-jbc+1)
        yn=wseal + ly * float(j-jbc+1)/float(jec-jbc+1)
         ys=ys/toth
         yn=yn/toth
        q = keff *((t(iec+1,j) - t(iec,j))/(dx1+dx2)) - sp(isurf)
        if(q.gt.qmax) qmax=q
        if(ioflg.ne.1) goto 20
        write(iwq,66) ys,q
        write(iwq,66) yn,q
c         if((j.eq.jbcg-1).or.(j.eq.jecg)) then
c         write(iwq,66) yn, q+5.
c         write(iwq,66) yn, q-5.
c         write(iwq,66) yn,q
c         endif
20    continue
c  Adjacent top seal
      keff=fkeff(kseal,kg,dx1,dx2)
      do 30 j=jec+1,je
        ys=(float(j-jec-1)/float(je-jec))*wseal + wseal + ly
        yn=(float(j-jec  )/float(je-jec))*wseal + wseal + ly
         ys=ys/toth
         yn=yn/toth
        q = keff * (t(iec+1,j) - t(iec,j))/(dx1+dx2)
        if(q.gt.qmax) qmax=q
        if(ioflg.ne.1) goto 30
        write(iwq,66) ys,q
        write(iwq,66) yn,q
30    continue
      write(iwq,66) yn,zero
      ys=wseal
       ys=ys/toth
      write(iwq,66) ys,zero
c  Radiation from east cavity wall
      do 40 j=jbc,jec
        isurf=rcg
        if(j.gt.jecg) isurf=rcg - (j-jecg)
        if(j.lt.jbcg) isurf=rcg + (jbcg-j)
        ys=wseal + ly * float(j-jbc)/float(jec-jbc+1)
        yn=wseal + ly * float(j-jbc+1)/float(jec-jbc+1)
         ys=ys/toth
         yn=yn/toth
        q =-1.* sp(isurf)
        if(ioflg.ne.1) goto 40
        write(iwq,66) ys,q
        write(iwq,66) yn,q
          if((j.eq.jbcg-1).or.(j.eq.jecg)) then
          write(iwq,66) yn, q+3.
          write(iwq,66) yn, q-3.
          write(iwq,66) yn,q
          endif
40      continue
      write(iwq,66) yn,zero
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
c  This subroutine records the heat flux [W/sqm] distribution as a
c  function of y (the vertical dimension) at the east face of the
c  right hand sheet of glass.
c
      subroutine dummy(t,twe,tww,qmax,ibc,iec,jbc,jec,
     &                  ly,wseal,kg,kseal,con,kn ,tg,diep,
     &                  ib,ie,jb,je,id,jd,ioflg,iwq)
c
      real ly,kg,kseal,keff,kn
      real t(42,302)
66    format(2x,f8.4,2x,f8.1)
      qmax=0.
      dx1=tg/2.
      dx2=tn/2.
c  Adjacent bottom seal
      keff=fkeff(kg,kn,dx1,dx2)
      do 10 j=jb,jbc-1
        ys=(float(j-jb)/float(jbc-jb))*wseal
        yn=(float(j-jb+1)/float(jbc-jb))*wseal
        q = keff * (t(ie,j) - t(ie-1,j))/(dx1+dx2)
        if(q.gt.qmax) qmax=q
        if(ioflg.ne.1) goto 10
        write(iwq,66) ys,q
        write(iwq,66) yn,q
10    continue
c  Adjacent cavity
      do 20 j=jbc,jec
        ys=wseal + ly * float(j-jbc)/float(jec-jbc+1)
        yn=wseal + ly * float(j-jbc+1)/float(jec-jbc+1)
        q = keff * (t(ie,j) - t(ie-1,j))/(dx1+dx2)
        if(q.gt.qmax) qmax=q
        if(ioflg.ne.1) goto 20
        write(iwq,66) ys,q
        write(iwq,66) yn,q
20    continue
c  Adjacent top seal
```

```
        do 30 j=jec+1,je
          ys=(float(j-jec-1)/float(je-jec))*wseal + wseal + ly
          yn=(float(j-jec  )/float(je-jec))*wseal + wseal + ly
          q = keff * (t(ie,j) - t(ie-1,j))/(dx1+dx2)
          if(q.gt.qmax) qmax=q
          if(ioflg.ne.1) goto 30
          write(iwq,66) ys,q
          write(iwq,66) yn,q
 30     continue
        return
        end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
        function fkeff(k1,k2,d1,d2)
        real k1,k2
        denom=(d1/k1)+(d2/k2)
        fkeff=(d1+d2)/denom
        return
        end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
        subroutine nulnor(dnt,ib,ie,jb,je)
        real dnt(42,302)
        do 10 i=ib,ie
          do 20 j=jb-1,je
            dnt(i,j)=1.e-10
 20       continue
 10     continue
        return
        end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
c  calcqb calculates the average heat flux, qbar, at all possible
c  vertical cuts through the igu problem domain.  Other quantities
c  returned include the max, min and east wall values of qbar as well
c  as the average of all values of qbar (i.e., q-bar-bar).
c
        subroutine calcqb(qbbar,qbmax,qbmin,qbeast,rmep,det,t,u,tww,twe,
     &                    con,den,cp,
     &                    kseal,kg,kn,
     &                    diep,djnp,tg,tn,dyseal,ly,wseal,
     &                    sp,jbcg,jecg,lcg,rcg,tl,tr,br,nsurf,
     &                    ibc,iec,jbc,jec,
     &                    ib,ie,jb,je,id,jd, docds,oput,idbg)
c
        real  rmep(42,302), det(42,302), alpha, beta
        real t(42,302), u(42,302), kseal, kg, kn, keff
        real ly
        real sp(200)
        integer lcg,rcg,tl,tr,br,nsurf
        integer oput
        higu=ly + 2.*wseal
        tref=0.
        qbbar=0.
```

```
      qbmax=-1000.
      qbmin=1000.
      if(idbg.eq.1) then
         write(oput,*) lcg,tl,tr,rcg,br,nsurf
         do 200 isurf=1,nsurf
            write(oput,*) isurf,sp(isurf)
200      continue
      endif
c  surface of west plate
        qb=0.
        do 30 j=jb,jbc-1
          qb = qb + dyseal*kn*(t(ib,j)-tww)/(tn/2.)
30      continue
        do 31 j=jbc,jec
          qb = qb + djnp*kn*(t(ib,j)-tww)/(tn/2.)
31      continue
        do 32 j=jec+1,je
          qb = qb + dyseal*kn*(t(ib,j)-tww)/(tn/2.)
32      continue
        qb=qb/higu
      if(idbg.eq.1) write(oput,*) ib-1,qb
        qbbar=qbbar+qb
        if(qb.gt.qbmax) qbmax=qb
        if(qb.lt.qbmin) qbmin=qb
c  west neoprene/glass interface
        keff=fkeff(kn,kg,tn,tg)
        qb=0.
        do 40 j=jb,jbc-1
          qb = qb + dyseal*keff*(t(ib+1,j)-t(ib,j))/((tn+tg)/2.)
40      continue
        do 41 j=jbc,jec
          qb = qb + djnp*keff*(t(ib+1,j)-t(ib,j))/((tn+tg)/2.)
41      continue
        do 42 j=jec+1,je
          qb = qb + dyseal*keff*(t(ib+1,j)-t(ib,j))/((tn+tg)/2.)
42      continue
        qb=qb/higu
      if(idbg.eq.1) write(oput,*) ib,qb
        qbbar=qbbar+qb
        if(qb.gt.qbmax) qbmax=qb
        if(qb.lt.qbmin) qbmin=qb
c
c  Calculate average radiative flux across vertical surface along
c  west cavity wall
c
      qbrad=0.
      do 110 isurf=1,lcg-1
        qbrad = qbrad + sp(isurf)*djnp
110   continue
      qbrad = qbrad + sp(lcg)*djnp*float(jecg-jbcg+1)
      do 120 isurf=lcg+1,tl-1
        qbrad = qbrad + sp(isurf)*djnp
120   continue
      qbrad = qbrad/higu
```

```
c
c
c  west glass/fill-gas surface
        qb=0.
        do 50 j=jb,jbc-1
          keff=fkeff(kseal,kg,diep,tg)
          qb = qb + dyseal*keff*(t(ibc,j)-t(ibc-1,j))/((diep+tg)/2.)
50      continue
        do 51 j=jbc,jec
          keff=fkeff(con,kg,diep,tg)
          qb = qb + djnp*keff*(t(ibc,j)-t(ibc-1,j))/((diep+tg)/2.)
51      continue
        do 52 j=jec+1,je
          keff=fkeff(kseal,kg,diep,tg)
          qb = qb + dyseal*keff*(t(ibc,j)-t(ibc-1,j))/((diep+tg)/2.)
52      continue
        qb= qbrad + qb/higu
        qbbar=qbbar+qb
      if(idbg.eq.1) write(oput,*) ib+1,qb
        if(qb.gt.qbmax) qbmax=qb
        if(qb.lt.qbmin) qbmin=qb
c  surfaces in fill gas cavity
      do 10 i=ibc,iec-1
        qb=0.
        do 60 j=jb,jbc-1
          qb = qb + dyseal*kseal*(t(i+1,j)-t(i,j))/diep
60      continue
        do 61 j=jbc,jec
          pe=rmep(i,j)/det(i,j)
          call prfl(alpha,beta,pe, docds)
          tfe=(0.5+alpha)*t(i,j) + (0.5-alpha)*t(i+1,j)
          qb=qb + djnp*(beta*con*(t(i+1,j)-t(i,j))/diep
     &            - u(i,j) *cp * den *(tfe-tref))
61      continue
        do 62 j=jec+1,je
          qb = qb + dyseal*kseal*(t(i+1,j)-t(i,j))/diep
62      continue
              isn = tl + (i-ibc)
              iss = nsurf - (i-ibc)
              qbrad = qbrad + (sp(isn)+sp(iss))*diep/higu
        qb= qbrad + qb/higu
        qbbar=qbbar+qb
      if(idbg.eq.1) write(oput,*) i,qb
        if(qb.gt.qbmax) qbmax=qb
        if(qb.lt.qbmin) qbmin=qb
10    continue
c  east glass/fill-gas surface
        qb=0.
        do 70 j=jb,jbc-1
          keff=fkeff(kseal,kg,diep,tg)
          qb = qb + dyseal*keff*(t(iec+1,j)-t(iec,j))/((diep+tg)/2.)
70      continue
        do 71 j=jbc,jec
          keff=fkeff(con,kg,diep,tg)
          qb = qb + djnp*keff*(t(iec+1,j)-t(iec,j))/((diep+tg)/2.)
71      continue
        do 72 j=jec+1,je
          keff=fkeff(kseal,kg,diep,tg)
          qb = qb + dyseal*keff*(t(iec+1,j)-t(iec,j))/((diep+tg)/2.)
72      continue
              qbrad = qbrad + (sp(tr)+sp(br))*diep/higu
        qb= qbrad + qb/higu
        qbbar=qbbar+qb
      if(idbg.eq.1) write(oput,*) iec,qb
        if(qb.gt.qbmax) qbmax=qb
        if(qb.lt.qbmin) qbmin=qb
c  east neoprene/glass interface
        keff=fkeff(kn,kg,tn,tg)
        qb=0.
        do 80 j=jb,jbc-1
          qb = qb + dyseal*keff*(t(ie,j)-t(ie-1,j))/((tn+tg)/2.)
80      continue
        do 81 j=jbc,jec
          qb = qb + djnp*keff*(t(ie,j)-t(ie-1,j))/((tn+tg)/2.)
81      continue
        do 82 j=jec+1,je
          qb = qb + dyseal*keff*(t(ie,j)-t(ie-1,j))/((tn+tg)/2.)
82      continue
        qb=qb/higu
      if(idbg.eq.1) write(oput,*) iec+1,qb
        qbbar=qbbar+qb
        if(qb.gt.qbmax) qbmax=qb
        if(qb.lt.qbmin) qbmin=qb
c  surface of east plate
        qb=0.
        do 90 j=jb,jbc-1
          qb = qb + dyseal*kn*(twe-t(ie,j))/(tn/2.)
90      continue
        do 91 j=jbc,jec
          qb = qb + djnp*kn*(twe-t(ie,j))/(tn/2.)
91      continue
        do 92 j=jec+1,je
          qb = qb + dyseal*kn*(twe-t(ie,j))/(tn/2.)
92      continue
        qb=qb/higu
      if(idbg.eq.1) write(oput,*) ie,qb
        qbbar=qbbar+qb
        if(qb.gt.qbmax) qbmax=qb
        if(qb.lt.qbmin) qbmin=qb

      qbbar=qbbar/float(ie-ib+2)
      qbeast=qb
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
      subroutine ftofy(t,tmin,tmax,iwty,ioflg,icut,
     &                 ly,wseal,
     &                 ibc,iec,jbc,jec,jbcg,jecg,
     &                 ib,ie,jb,je,id,jd)
      real ly
      real t(42,302)
 66   format(2x,f8.4,2x,f8.2)
      toth=ly + 2.*wseal
      tmax=-1000.
      tmin=1000.
c  Adjacent bottom seal
      do 10 j=jb,jbc-1
        ys=(float(j-jb)/float(jbc-jb))*wseal
        yn=(float(j-jb+1)/float(jbc-jb))*wseal
         ys=ys/toth
         yn=yn/toth
        if(t(icut,j).gt.tmax) tmax=t(icut,j)
        if(t(icut,j).lt.tmin) tmin=t(icut,j)
        if(ioflg.ne.1) goto 10
        write(iwty,66) ys,t(icut,j)
        write(iwty,66) yn,t(icut,j)
 10   continue
      if(ioflg.ne.1) goto 11
      write(iwty,66) yn+0.1,t(icut,jbc-1)-0.4
      write(iwty,66) yn,t(icut,jbc-1)
c  Adjacent cavity
 11   do 20 j=jbc,jec
        ys=wseal + ly * float(j-jbc)/float(jec-jbc+1)
        yn=wseal + ly * float(j-jbc+1)/float(jec-jbc+1)
         ys=ys/toth
         yn=yn/toth
        if(t(icut,j).gt.tmax) tmax=t(icut,j)
        if(t(icut,j).lt.tmin) tmin=t(icut,j)
        if(ioflg.ne.1) goto 20
        write(iwty,66) ys,t(icut,j)
        write(iwty,66) yn,t(icut,j)
c         if((j.eq.jbcg-1).or.(j.eq.jecg)) then
c         write(iwty,66) yn, t(icut,j)+0.1
c         write(iwty,66) yn, t(icut,j)-0.1
c         write(iwty,66) yn, t(icut,j)
c         endif
 20   continue
c  Adjacent top seal
      do 30 j=jec+1,je
        ys=(float(j-jec-1)/float(je-jec))*wseal + wseal + ly
        yn=(float(j-jec  )/float(je-jec))*wseal + wseal + ly
         ys=ys/toth
         yn=yn/toth
        if(t(icut,j).gt.tmax) tmax=t(icut,j)
        if(t(icut,j).lt.tmin) tmin=t(icut,j)
        if(ioflg.ne.1) goto 30
        write(iwty,66) ys,t(icut,j)
          if(j.eq.jec+1) then
          write(iwty,66) ys-0.1,t(icut,j)-0.4
          write(iwty,66) ys,t(icut,j)
          endif
        write(iwty,66) yn,t(icut,j)
```

```
30    continue
99    return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
c  This subroutine records the heat flux distribution as a
c  function of y (the vertical dimension) at the east (isothermal)
c  wall of the problem domain (i.e., copper/neoprene interface).
c
      subroutine fqep(t,twe,tww,iwq,qshew,qelsh,qmax,
     &                 kn,tn,jbc,jec,ly,wseal,jbcg,jecg,
     &                 ib,ie,jb,je,id,jd)
c
      real t(42,302)
      real qshew(5), qelsh(3)
      real yshews(5), yshewn(5), yelshs(3), yelshn(3)
      real kn, ly
      zero=0.
66    format(2x,f7.4,2x,f8.1)
c
      yelshs(1)=0.5/25.
      yelshn(1)=(0.5+(23.5/3.))/25.
      yelshs(2)=(0.5+(23.5/3.)+0.25)/25.
      yelshn(2)=(0.5+2.*(23.5/3.)+0.25)/25.
      yelshs(3)=(0.5+2.*(23.5/3.)+0.25+0.25)/25.
      yelshn(3)=(0.5+3.*(23.5/3.)+0.25+0.25)/25.
c
      topbot=3./16.
      hplt=5.5
      tothgt=29. + (9./16.)
      spcbet=(tothgt -5.*hplt -2.*topbot)/4.
      yshews(1)=(topbot + 0.*hplt + 0.*spcbet)/tothgt
      yshewn(1)=(topbot + 1.*hplt + 0.*spcbet)/tothgt
      yshews(2)=(topbot + 1.*hplt + 1.*spcbet)/tothgt
      yshewn(2)=(topbot + 2.*hplt + 1.*spcbet)/tothgt
      yshews(3)=(topbot + 2.*hplt + 2.*spcbet)/tothgt
      yshewn(3)=(topbot + 3.*hplt + 2.*spcbet)/tothgt
      yshews(4)=(topbot + 3.*hplt + 3.*spcbet)/tothgt
      yshewn(4)=(topbot + 4.*hplt + 3.*spcbet)/tothgt
      yshews(5)=(topbot + 4.*hplt + 4.*spcbet)/tothgt
      yshewn(5)=(topbot + 5.*hplt + 4.*spcbet)/tothgt
c
      nx=ie-ib+1
      ny=je-jb+1
      do 12 i=1,3
        qelsh(i)=0.
12    continue
      do 13 i=1,5
        qshew(i)=0.
13    continue
      qmax=0.
      do 11 j=jb,je
c       calculate south and north limits of each wall element
        ys = wseal*float(j-jb)/float(jbc-jb)
        if(j.gt.jbc) ys=wseal+ly*float(j-jbc)/float(jec-jbc+1)
        if(j.gt.jec+1) ys=wseal+ly+wseal*float(j-jec-1)/float(je-jec)
        yn=ys + wseal/float(jbc-jb)
        if(j.ge.jbc) yn=ys+ly/float(jec-jbc+1)
        if(j.gt.jec) yn=ys+wseal/float(je-jec)
        ys=ys/(wseal+ly+wseal)
        yn=yn/(wseal+ly+wseal)
         q = kn*(twe-t(ie,j))/(tn/2.)
        if(q.gt.qmax) qmax=q
c
        write(iwq,66) ys,q
          if(j.eq.jec+1) then
          write(iwq,66) ys-0.1, q+30.
          write(iwq,66) ys,q
          endif
        write(iwq,66) yn,q
          if(j.eq.jbc-1) then
          write(iwq,66) yn+0.1, q+30.
          write(iwq,66) yn,q
          endif
c         if((j.eq.jbcg-1).or.(j.eq.jecg)) then
c         write(iwq,66) yn, q+5.
c         write(iwq,66) yn, q-5.
c         write(iwq,66) yn,q
c         endif
c
c       Figure out which heater plate this wall element sits in
        do 20 i=1,3
          ipelsh=i
          if((yn.gt.yelshs(i)).and.(ys.lt.yelshn(i))) goto 21
          ipelsh=0
20      continue
21      do 30 i=1,5
          ipshew=i
          if((yn.gt.yshews(i)).and.(ys.lt.yshewn(i))) goto 31
          ipshew=0
30      continue
c       Calculate heat flux (average) over each guarded heater plate
c       in ElSherbiny's 3 plate rig and Shewen's 5 plate rig
31      if(ipelsh.eq.0) goto 55
c       Figure out overlap between wall element and heater plate
        olap = yn-ys
        if(ys.lt.yelshs(ipelsh)) olap=yn-yelshs(ipelsh)
        if(yn.gt.yelshn(ipelsh)) olap=yelshn(ipelsh)-ys
        qelsh(ipelsh)=qelsh(ipelsh)+q*olap/(yelshn(ipelsh) -
     &                                        yelshs(ipelsh))
c       Repeat for Shewen's plates
55      if(ipshew.eq.0) goto 11
        olap = yn-ys
        if(ys.lt.yshews(ipshew)) olap=yn-yshews(ipshew)
        if(yn.gt.yshewn(ipshew)) olap=yshewn(ipshew)-ys
        qshew(ipshew)=qshew(ipshew)+q*olap/(yshewn(ipshew) -
     &                                      yshews(ipshew))
11    continue
```

```
      write(iwq,66) yn,zero
c     Write heat flux averages over guarded heater
c     plate locations to grapher file.
c
c     write(iwq,*)'     ******************'
      do 40 i=1,3
        write(iwq,66) yelshs(i), zero
        write(iwq,66) yelshs(i), qelsh(i)
        write(iwq,66) yelshn(i), qelsh(i)
        write(iwq,66) yelshn(i), zero
40    continue
c
          if(2.gt.1) goto 99
c
c     Repeat for Shewen's heater plate rig
c     write(iwq,*)'     ******************'
      do 80 i=1,5
        write(iwq,66) yshews(i), zero
        write(iwq,66) yshews(i), qshew(i)
        write(iwq,66) yshewn(i), qshew(i)
        write(iwq,66) yshewn(i), zero
80    continue
99    return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
c  This subroutine estimates the temperature difference between
c  the vertical walls of the cavity for the purpose of calculating
c  a mean gas temperature and in turn for calculating Ra.
c
      subroutine tcav(t,tm,deltat,
     &                con,diep,kg,tg,
     &                ibc,iec,jbc,jec)
      real t(42,302), kg
      denom=(con/diep)+(kg/tg)
      jms=ifix((float(jbc+jec)/2.)+0.1)
      jmn=jms
      ndivy=jec-jbc+1
      itest=ndivy - 2 * ifix(0.1+float(ndivy)/2.)
      if(itest.eq.0) jmn=jmn+1
      twwn=(t(ibc-1,jmn)*(kg/tg)+t(ibc,jmn)*(con/diep))/denom
      twws=(t(ibc-1,jms)*(kg/tg)+t(ibc,jms)*(con/diep))/denom
      tww=(twwn+twws)/2.
      twen=(t(iec+1,jmn)*(kg/tg)+t(iec,jmn)*(con/diep))/denom
      twes=(t(iec+1,jms)*(kg/tg)+t(iec,jms)*(con/diep))/denom
      twe=(twen+twes)/2.
      deltat=twe-tww
      tm=(twe+tww)/2.
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine spscan(sp,i1,i2,spmax,spmin)
      real sp(200)
      spmax=-1000.
      spmin=1000.
      do 10 isurf=i1,i2
        if(sp(isurf).gt.spmax) spmax=sp(isurf)
        if(sp(isurf).lt.spmin) spmin=sp(isurf)
10    continue
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      function fqrld(tl,tr,epsl,epsr)
      sig=5.6693
      f=sig/((1./epsl)+(1./epsr)-1.)
      fqrld=f * ((tr/100.)**4 - (tl/100.)**4)
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      function fcgc(qelsh,deltat,tg,kg)
      real qelsh(3),kg
      dtg=qelsh(2)*tg/kg
      fcgc=qelsh(2)/(deltat + 2.*dtg)
      return
      end
```

```
c
c                   IGUSUB.FOR              John L. Wright
c
c      2-D Heat Transfer in a Integrated Glazing Unit
c
c                                           August, 1989
c
c                    Subroutine Source Code
c
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C
      subroutine coeff(ap,ae,aw,an,as,bp,r,q,phi,
     &                 de,dn,rmep,rmnp,alphae,alphan,betae,betan,e,
     &                 ib,ie,jb,je,id,jd)
C
      real r(42,302), q(42,302)
      real ae(42,302), aw(42,302), an(42,302),as(42,302)
      real ap(42,302), bp(42,302)
      real phi(42,302)
      real de(42,302), dn(42,302)
      real rmep(42,302),rmnp(42,302)
      real alphae(42,302), alphan(42,302), betae(42,302), betan(42,302)
      integer ib,ie,jb,je,id,jd
      do 10 i=ib,ie
        do 20 j=jb,je
          ae(i,j)=de(i,j)*betae(i,j) -0.5*rmep(i,j) +
     &             abs(rmep(i,j))*alphae(i,j)
          aw(i,j)=de(i-1,j)*betae(i-1,j) + 0.5*rmep(i-1,j) +
     &             abs(rmep(i-1,j))*alphae(i-1,j)
          an(i,j)=dn(i,j)*betan(i,j) - 0.5*rmnp(i,j) +
     &             abs(rmnp(i,j))*alphan(i,j)
          as(i,j)=dn(i,j-1)*betan(i,j-1) +0.5*rmnp(i,j-1) +
     &             abs(rmnp(i,j-1))*alphan(i,j-1)
          sumnab  =ae(i,j)+aw(i,j)+an(i,j)+as(i,j)
          ap(i,j)=(sumnab-r(i,j))*(1.+(1./e))
          bp(i,j)=q(i,j)+(sumnab-r(i,j))*phi(i,j)/e
20      continue
10    continue
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C
      subroutine coefp(app,ape,apw,apn,aps,
     &                 aup,aue,auw,aun,aus,
     &                 avp,ave,avw,avn,avs,
     &                 arep,arnp,den,
     &                 depp, dnpp,
     &                 ib,ie,jb,je,id,jd)
C
      real ape(42,302), apw(42,302), apn(42,302),aps(42,302),app(42,302)
      real aue(42,302), auw(42,302), aun(42,302),aus(42,302),aup(42,302)
      real ave(42,302), avw(42,302), avn(42,302),avs(42,302),avp(42,302)
      real depp(42,302), dnpp(42,302)
      integer ib,ie,jb,je,id,jd
```

```
c
c  calculate e/w diffusion coeficients for p' at all interior borders
      call null(depp,ib,ie,jb,je,id,jd)
      do 30 i=ib,ie-1
        do 40 j=jb,je
          sum=auw(i,j)+aue(i,j)+aun(i,j)+aus(i,j)
          depp(i,j)=arep/(aup(i,j)-sum)
40      continue
30    continue
c
c  calculate n/s diffusion coeficients for p' at all interior borders
      call null(dnpp,ib,ie,jb,je,id,jd)
      do 70 i=ib,ie
        do 80 j=jb,je-1
          sum=avw(i,j)+ave(i,j)+avn(i,j)+avs(i,j)
          dnpp(i,j)=arnp/(avp(i,j)-sum)
80      continue
70    continue
c
c  calculate coefficients for p' equation
      call null(ape,ib,ie,jb,je,id,jd)
      call null(apw,ib,ie,jb,je,id,jd)
      call null(apn,ib,ie,jb,je,id,jd)
      call null(aps,ib,ie,jb,je,id,jd)
      do 10 i=ib,ie
        do 20 j=jb,je
          ape(i,j)=den * arep * depp(i,j)
          apw(i,j)=den * arep * depp(i-1,j)
          apn(i,j)=den * arnp * dnpp(i,j)
          aps(i,j)=den * arnp * dnpp(i,j-1)
20      continue
10    continue
c
C  Boundary condition at east edge
      do 110 j=jb,je
        apbc=aup(ie,j)
        awbc=auw(ie,j)
c
        ape(ie,j)=0.
        apw(ie,j)=apw(ie,j)-depp(ie-1,j)*den*arep*awbc/apbc
110   continue
c
C  Boundary condition at north edge
      do 120 i=ib,ie
        apbc=avp(i,je)
        asbc=avs(i,je)
c
        apn(i,je)=0.
        aps(i,je)=aps(i,je)-dnpp(i,je-1)*den*arnp*asbc/apbc
120   continue
c
C  Boundary condition at west edge
      do 130 j=jb,je
        apbc=aup(ib-1,j)
```

```
         aebc=aue(ib-1,j)
c
         apw(ib,j)=0.
         ape(ib,j)=ape(ib,j)-depp(ib,j)*den*arep*aebc/apbc
130    continue
c
C  Boundary condition at south edge
       do 140 i=ib,ie
         apbc=avp(i,jb-1)
         anbc=avn(i,jb-1)
c
         aps(i,jb)=0.
         apn(i,jb)=apn(i,jb)-dnpp(i,jb)*den*arnp*anbc/apbc
140    continue
c
       do 50 i=ib,ie
         do 60 j=jb,je
           app(i,j)  =ape(i,j)+apw(i,j)+apn(i,j)+aps(i,j)
60       continue
50     continue
       return
       end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C
       subroutine sorcp(bp,rmep,rmnp,
     &                  ib,ie,jb,je,id,jd)
C
       real rmep(42,302), rmnp(42,302)
       real bp(42,302)
c  bp is the net influx of mass at volume i,j
       do 10 i=ib,ie
         do 20 j=jb,je
           bp(i,j)=rmep(i-1,j)+rmnp(i,j-1)-rmep(i,j)-rmnp(i,j)
20       continue
10     continue
       return
       end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C
       subroutine sorcu(ru,qu,
     &                  ib,ie,jb,je,id,jd)
C
       real ru(42,302), qu(42,302)
       do 10 i=ib,ie-1
         do 20 j=jb,je
           ru(i,j)=0.
           qu(i,j)=0.
20       continue
10     continue
       return
       end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C
       subroutine sorcv(rv,qv,
     &                 ib,ie,jb,je,id,jd)
c
      real rv(42,302), qv(42,302)
      do 10 i=ib,ie
        do 20 j=jb,je-1
          rv(i,j)=0.
          qv(i,j)=0.
20      continue
10    continue
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine sorcpu(bu,p,area,
     &                 ib,ie,jb,je,id,jd)
c
      real bu(42,302), p(42,302)
      do 10 i=ib,ie-1
        do 20 j=jb,je
          bu(i,j)=bu(i,j)+area*(p(i,j)-p(i+1,j))
20      continue
10    continue
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine sorcpv(bv,p,area,
     &                 ib,ie,jb,je,id,jd)
c
      real bv(42,302), p(42,302)
      do 10 i=ib,ie
        do 20 j=jb,je-1
          bv(i,j)=bv(i,j)+area*(p(i,j)-p(i,j+1))
20      continue
10    continue
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine dt(det,dnt,con,diep,djnp,arep,arnp,cp,
     &              ib,ie,jb,je,id,jd)
c
      real det(42,302), dnt(42,302), con, diep, djnp, arep, arnp
      real cp
      integer ib,ie,jb,je,id,jd
c
      call null(det,ib,ie,jb,je,id,jd)
c  Use the first of the two following lines if diffusion terms are
c  to be calculated at the cavity walls.
c     do 10 i=ib-1,ie
      do 10 i=ib  ,ie-1
        do 20 j=jb,je
          det(i,j)=con*arep/(cp*diep)
20      continue
```

```
10    continue
c
      call null(dnt,ib,ie,jb,je,id,jd)
      do 30 i=ib,ie
c  Use the first of the two following lines if diffusion terms are
c  to be calculated at the cavity walls.
c       do 40 j=jb-1,je
        do 40 j=jb  ,je-1
          dnt(i,j)=con*arnp/(cp*djnp)
40      continue
30    continue
c
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C
      subroutine dtsol(det,dnt,con,diep,djnp,arep,arnp,cp,
     &           dyseal,kseal,kg,tg,kn,tn,
     &           ibc,iec,jbc,jec,
     &           ib,ie,jb,je,id,jd)
C
c  All interior faces of cavity control volumes should already
c  have been accounted for in subroutine dt(det,dnt,.......)
c
      real kseal, kg, kn, keff
      real det(42,302), dnt(42,302), con, diep, djnp, arep, arnp
      real cp
      integer ib,ie,jb,je,id,jd
c
c  Specific heat values for neoprene, glass and seal set equal to
c  specific heat of fill gas since no convection takes place in
c  solid materials and only the steady state solution is sought
      cpn=cp
      cpg=cp
      cpseal=cp
      cpeff=cp
c
c     east faces in contact with isothermal copper plates
      do 10 i=ib-1,ie,ie-ib+1
c       adjacent cavity
        do 20 j=jbc,jec
          det(i,j)=kn*arep/(cpn*tn)
20      continue
c       adjacent bottom seal
        do 21 j=jb,jbc-1
          det(i,j)=kn*dyseal/(cpn*tn)
21      continue
c       adjacent top seal
        do 22 j=jec+1,je
          det(i,j)=kn*dyseal/(cpn*tn)
22      continue
10    continue
c
c     east faces at glass/neoprene interface
```

```
      do 35 i=ib,ie-1,ie-ib-1
       keff = fkeff(kg,kn,tg,tn)
c      adjacent cavity
       do 30 j=jbc,jec
        det(i,j)= keff*arep/(cpeff*((tn+tg)/2.))
30     continue
c      adjacent bottom seal
       do 31 j=jb,jbc-1
        det(i,j)= keff*dyseal/(cpeff*((tn+tg)/2.))
31     continue
c      adjacent top seal
       do 32 j=jec+1,je
        det(i,j)= keff*dyseal/(cpeff*((tn+tg)/2.))
32     continue
35    continue
c
c     east faces at glass/cavity and glass/seal interfaces
      do 75 i=ibc-1,iec,iec-ibc+1
c      adjacent cavity
       keff = fkeff(kg,con,tg,diep)
       do 70 j=jbc,jec
        det(i,j)= keff*arep/(cpeff*((diep+tg)/2.))
70     continue
c      adjacent bottom seal
       keff = fkeff(kg,kseal,tg,diep)
       do 71 j=jb,jbc-1
        det(i,j)= keff*dyseal/(cpeff*((diep+tg)/2.))
71     continue
c      adjacent top seal
       do 72 j=jec+1,je
        det(i,j)= keff*dyseal/(cpeff*((diep+tg)/2.))
72     continue
75    continue
c
c      north faces in neoprene and glass layers including faces
c      next to fictitous volumes but excluding faces between
c      volumes where volume size changes
c
c       adjacent bottom seal
       do 40 j=jb-1,jbc-2
        dnt(ib  ,j)=kn*tn/(cpn*dyseal)
        dnt(ib+1,j)=kg*tg/(cpg*dyseal)
        dnt(ie-1,j)=kg*tg/(cpg*dyseal)
        dnt(ie  ,j)=kn*tn/(cpn*dyseal)
40     continue
c       adjacent cavity
       do 41 j=jbc,jec-1
        dnt(ib  ,j)=kn*tn/(cpn*djnp)
        dnt(ib+1,j)=kg*tg/(cpg*djnp)
        dnt(ie-1,j)=kg*tg/(cpg*djnp)
        dnt(ie  ,j)=kn*tn/(cpn*djnp)
41     continue
c       adjacent top seal
       do 42 j=jec+1,je
```

```
          dnt(ib  ,j)=kn*tn/(cpn*dyseal)
          dnt(ib+1,j)=kg*tg/(cpg*dyseal)
          dnt(ie-1,j)=kg*tg/(cpg*dyseal)
          dnt(ie  ,j)=kn*tn/(cpn*dyseal)
42      continue
c
c       interior and bc north faces for seals
        do 55 i=ibc,iec
c         bottom seal
          do 50 j=jb-1,jbc-2
            dnt(i,j)=kseal*arnp/(cpseal*dyseal)
50        continue
c         top seal
          do 52 j=jec+1,je
            dnt(i,j)=kseal*arnp/(cpseal*dyseal)
52        continue
55      continue
c
c       interior east faces of seal volumes
        do 65 i=ibc,iec-1
c         bottom seal
          do 60 j=jb  ,jbc-1
            det(i,j)=kseal*dyseal/(cpseal*diep)
60        continue
c         top seal
          do 62 j=jec+1,je
            det(i,j)=kseal*dyseal/(cpseal*diep)
62        continue
65      continue
c
c     all north faces at lines along top and bottom surfaces of
c     cavity
      do 95 j=jbc-1,jec,jec+1-jbc
c       adjacent cavity
        keff = fkeff(kseal,con,dyseal,djnp)
        do 90 i=ibc,iec
          dnt(i,j)= keff*arnp/(cpeff*((djnp+dyseal)/2.))
90      continue
c       in glass sections
        do 91 i=ibc-1,iec+1,iec+1-ibc+1
          dnt(i,j)= kg*tg/(cpeff*((djnp+dyseal)/2.))
91      continue
c       in neoprene sections
        do 92 i=ib,ie,ie-ib
          dnt(i,j)= kn*tn/(cpeff*((djnp+dyseal)/2.))
92      continue
95    continue
c
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C
      subroutine du(deu,dnu,visc,dieu,djnu,areu,arnu,
     &               ib,ie,jb,je,id,jd)
```

```
c
      real deu(42,302), dnu(42,302)
      integer ib,ie,jb,je,id,jd
c
      call null(deu,ib,ie,jb,je,id,jd)
      do 10 i=ib-1,ie-1
        do 20 j=jb,je
          deu(i,j)=visc*areu/dieu
20      continue
10    continue
c
      call null(dnu,ib,ie,jb,je,id,jd)
      do 30 i=ib,ie-1
        do 40 j=jb-1,je
          dnu(i,j)=visc*arnu/djnu
40      continue
30    continue
c
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine dv(dev,dnv,visc,diev,djnv,arev,arnv,
     &              ib,ie,jb,je,id,jd)
c
      real dev(42,302), dnv(42,302)
      integer ib,ie,jb,je,id,jd
c
      call null(dev,ib,ie,jb,je,id,jd)
      do 10 i=ib-1,ie
        do 20 j=jb,je-1
          dev(i,j)=visc*arev/diev
20      continue
10    continue
c
      call null(dnv,ib,ie,jb,je,id,jd)
      do 30 i=ib,ie
        do 40 j=jb-1,je-1
          dnv(i,j)=visc*arnv/djnv
40      continue
30    continue
c
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine bndct(atp,ate,atw,atn,ats,btp,
     &                 alphae,alphan,betae,betan,tww,twe,
     &                 ib,ie,jb,je,id,jd)
c
      real ate(42,302), atw(42,302), atn(42,302),ats(42,302)
      real atp(42,302), btp(42,302)
      real alphae(42,302), alphan(42,302), betae(42,302), betan(42,302)
      integer ib,ie,jb,je,id,jd
```

```
c
C  west side boundary condition, at x=0 T=Tb=tww
      tb=tww
      do 10 j=jb,je
        atp(ib-1,j)=1. + 2.*alphae(ib-1,j)
        ate(ib-1,j)=-1. +2.*alphae(ib-1,j)
        atw(ib-1,j)=0.
        ats(ib-1,j)=0.
        atn(ib-1,j)=0.
        btp(ib-1,j)=2.*tb
10    continue
c
C  east side boundary condition, at x=Lx T=Tb=twe
      tb=twe
      do 20 j=jb,je
        atp(ie+1,j)=1. - 2.*alphae(ie,j)
        ate(ie+1,j)=0.
        atw(ie+1,j)=-1. - 2.*alphae(ie,j)
        ats(ie+1,j)=0.
        atn(ie+1,j)=0.
        btp(ie+1,j)=2.*tb
20    continue
c
C  south side boundary condition, at y=0 dT/dy=0
      tb=0.
      do 30 i=ib,ie
        atp(i,jb-1)=1.
        ate(i,jb-1)=0.
        atw(i,jb-1)=0.
        ats(i,jb-1)=0.
        atn(i,jb-1)=1.
        btp(i,jb-1)=0.
30    continue
c
C  north side boundary condition, at y=Ly dT/dy=0
      tb=1.
      do 40 i=ib,ie
        atp(i,je+1)=1.
        ate(i,je+1)=0.
        atw(i,je+1)=0.
        ats(i,je+1)=1.
        atn(i,je+1)=0.
        btp(i,je+1)=0.
40    continue
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C
      subroutine bndcu(aup,aue,auw,aun,aus,bup,
     &                  alphae,alphan,betae,betan,
     &                  ib,ie,jb,je,id,jd)
C
      real aue(42,302), auw(42,302), aun(42,302),aus(42,302)
      real aup(42,302), bup(42,302)
      real alphae(42,302), alphan(42,302), betae(42,302), betan(42,302)
      integer ib,ie,jb,je,id,jd
c
c  west side boundary condition, at x=0 u=ub
      ub=0.0
      do 10 j=jb,je
        aup(ib-1,j)=1.
        aue(ib-1,j)=0.
        auw(ib-1,j)=0.
        aus(ib-1,j)=0.
        aun(ib-1,j)=0.
        bup(ib-1,j)=ub
10    continue
c
C  east side boundary condition, at x=Lx u=ub
      ub=0.
      do 20 j=jb,je
        aup(ie,j)=1.
        aue(ie,j)=0.
        auw(ie,j)=0.
        aus(ie,j)=0.
        aun(ie,j)=0.
        bup(ie,j)=ub
20    continue
c
C  south side boundary condition, at y=0 u=ub
      ub=0.
      do 30 i=ib,ie
        aup(i,jb-1)=1. + 2.*alphan(i,jb-1)
        aue(i,jb-1)=0.
        auw(i,jb-1)=0.
        aus(i,jb-1)=0.
        aun(i,jb-1)=-1. + 2.*alphan(i,jb-1)
        bup(i,jb-1)=2.*ub
30    continue
c
C  north side boundary condition, at y=Ly u=ub
      ub=0.
      do 40 i=ib,ie
        aup(i,je+1)=1. - 2.*alphan(i,je)
        aue(i,je+1)=0.
        auw(i,je+1)=0.
        aus(i,je+1)=-1. - 2.*alphan(i,je)
        aun(i,je+1)=0.
        bup(i,je+1)=2.*ub
40    continue
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C
      subroutine bndcv(avp,ave,avw,avn,avs,bvp,
     &                  alphae,alphan,betae,betan,
     &                  ib,ie,jb,je,id,jd)
C
```

```
      real ave(42,302), avw(42,302), avn(42,302),avs(42,302)
      real avp(42,302), bvp(42,302)
      real alphae(42,302), alphan(42,302), betae(42,302), betan(42,302)
      integer ib,ie,jb,je,id,jd
c
C  west side boundary condition, at x=0 v=vb
      vb=0.
      do 10 j=jb,je
        avp(ib-1,j)=1. + 2.*alphae(ib-1,j)
        ave(ib-1,j)=-1. +2.*alphae(ib-1,j)
        avw(ib-1,j)=0.
        avs(ib-1,j)=0.
        avn(ib-1,j)=0.
        bvp(ib-1,j)=2.*vb
10    continue
c
C  east side boundary condition, at x=Lx v=vb
      vb=0.
      do 20 j=jb,je
        avp(ie+1,j)=1. - 2.*alphae(ie,j)
        ave(ie+1,j)=0.
        avw(ie+1,j)=-1. - 2.*alphae(ie,j)
        avs(ie+1,j)=0.
        avn(ie+1,j)=0.
        bvp(ie+1,j)=2.*vb
20    continue
c
C  south side boundary condition, at y=0 v=vb
      vb=0.
      do 30 i=ib,ie
        avp(i,jb-1)=1.
        ave(i,jb-1)=0.
        avw(i,jb-1)=0.
        avs(i,jb-1)=0.
        avn(i,jb-1)=0.
        bvp(i,jb-1)=vb
30    continue
c
C  north side boundary condition, at y=Ly v=vb
      vb=0.
      do 40 i=ib,ie
        avp(i,je)=1.
        ave(i,je)=0.
        avw(i,je)=0.
        avs(i,je)=0.
        avn(i,je)=0.
        bvp(i,je)=vb
40    continue
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
C
      subroutine bcelim(atp,ate,atw,atn,ats,btp,
     &                  ireset,reset,phi,
     &                     ib,ie,jb,je,id,jd)
c
      real ate(42,302), atw(42,302), atn(42,302),ats(42,302)
      real atp(42,302), btp(42,302)
      real phi(42,302)
      integer ib,ie,jb,je,id,jd
c
c  absorb west side boundary condition
      do 10 j=jb,je
        atp(ib,j)=atp(ib,j)-atw(ib,j)*ate(ib-1,j)/atp(ib-1,j)
        btp(ib,j)=btp(ib,j)+atw(ib,j)*btp(ib-1,j)/atp(ib-1,j)
        atw(ib,j)=0.
        if(ireset.eq.1) phi(ib-1,j)=reset
10    continue
c
c  absorb east side boundary condition
      do 20 j=jb,je
        atp(ie,j)=atp(ie,j)-ate(ie,j)*atw(ie+1,j)/atp(ie+1,j)
        btp(ie,j)=btp(ie,j)+ate(ie,j)*btp(ie+1,j)/atp(ie+1,j)
        ate(ie,j)=0.
        if(ireset.eq.1) phi(ie+1,j)=reset
20    continue
c
c  absorb south side boundary condition
      do 30 i=ib,ie
        atp(i,jb)=atp(i,jb)-ats(i,jb)*atn(i,jb-1)/atp(i,jb-1)
        btp(i,jb)=btp(i,jb)+ats(i,jb)*btp(i,jb-1)/atp(i,jb-1)
        ats(i,jb)=0.
        if(ireset.eq.1) phi(i,jb-1)=reset
30    continue
c
c  absorb north side boundary condition
      do 40 i=ib,ie
        atp(i,je)=atp(i,je)-atn(i,je)*ats(i,je+1)/atp(i,je+1)
        btp(i,je)=btp(i,je)+atn(i,je)*btp(i,je+1)/atp(i,je+1)
        atn(i,je)=0.
        if(ireset.eq.1) phi(i,je+1)=reset
40    continue
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine sor2d(t,atp,ate,atw,atn,ats,btp,rlx,niter,
     &                     ib,ie,jb,je,id,jd)
c
      real ate(42,302), atw(42,302), atn(42,302),ats(42,302)
      real atp(42,302), btp(42,302)
      real t(42,302),rlx
      integer ib,ie,jb,je,id,jd
      integer niter
c
      do 5 iter=1,niter
c
        do 10 j=jb,je
```

```
            do 20 i=ib,ie
              dv=ate(i,j)*t(i+1,j)+atw(i,j)*t(i-1,j)
              dv=ats(i,j)*t(i,j-1)+atn(i,j)*t(i,j+1)+dv+btp(i,j)
              t(i,j)=(rlx/atp(i,j))*dv + (1.-rlx)*t(i,j)
20          continue
10        continue
c
5       continue
        return
        end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
        subroutine massp(rmep,rmnp,u,v,den,arep,arnp,
       &                  ib,ie,jb,je,id,jd,oput)
c
        real rmep(42,302),rmnp(42,302)
        real u(42,302), v(42,302)
        integer oput
c
        call null(rmep,ib,ie,jb,je,id,jd)
        do 10 j=jb,je
          do 20 i=ib-1,ie
            rmep(i,j)=u(i,j)*arep*den
20        continue
10      continue
c
        call null(rmnp,ib,ie,jb,je,id,jd)
        do 30 j=jb-1,je
          do 40 i=ib,ie
            rmnp(i,j)=v(i,j)*arnp*den
40        continue
30      continue
c
c  Check that mass flows at east and west cavity walls are
c  near zero and then manually set them exactly equal to zero
        do 50 j=jb,je
          do 60 i=ib-1,ie, ie-ib+1
            test=abs(rmep(i,j))
            if(test.gt.1.e-20) goto 88
            rmep(i,j) = 0.
60        continue
50      continue
c
c  Check that mass flows at north and south cavity walls are
c  near zero and then manually set them exactly equal to zero
        do 70 j=jb-1,je, je-jb+1
          do 80 i=ib,ie
            test=abs(rmnp(i,j))
            if(test.gt.1.e-20) goto 88
            rmnp(i,j)=0.
80        continue
70      continue
        goto 99
c  88 is an error condition where mass flow through the cavity
```

```
c  wall has been detected.
88    write(oput,*)'error in massp',i,j,rmep(i,j),rmnp(i,j)
      oput=0
99    return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine massu(rmeu,rmnu,rmep,rmnp,
     &                 ib,ie,jb,je,id,jd)
c
      real rmep(42,302),rmnp(42,302)
      real rmeu(42,302),rmnu(42,302)
c
      call null(rmeu,ib,ie,jb,je,id,jd)
      do 10 j=jb,je
        do 20 i=ib-1,ie-1
          rmeu(i,j)=(rmep(i,j)+rmep(i+1,j))/2.
20      continue
10    continue
c
      call null(rmnu,ib,ie,jb,je,id,jd)
      do 30 j=jb-1,je
        do 40 i=ib,ie-1
          rmnu(i,j)=(rmnp(i,j)+rmnp(i+1,j))/2.
40      continue
30    continue
c
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine massv(rmev,rmnv,rmep,rmnp,
     &                 ib,ie,jb,je,id,jd)
c
      real rmep(42,302),rmnp(42,302)
      real rmev(42,302),rmnv(42,302)
c
      call null(rmev,ib,ie,jb,je,id,jd)
      do 10 j=jb,je-1
        do 20 i=ib-1,ie
          rmev(i,j)=(rmep(i,j)+rmep(i,j+1))/2.
20      continue
10    continue
c
      call null(rmnv,ib,ie,jb,je,id,jd)
      do 30 j=jb-1,je-1
        do 40 i=ib,ie
          rmnv(i,j)=(rmnp(i,j)+rmnp(i,j+1))/2.
40      continue
30    continue
c
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine null(btp,ib,ie,jb,je,id,jd)
c
      real btp(42,302)
      integer ib,ie,jb,je,id,jd
      do 11 i=1,id
        do 21 j=1,jd
          btp(i,j)=0.
21      continue
11    continue
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine pshift(pp,ipref,jpref,ib,ie,jb,je,id,jd)
c
      real pp(42,302)
      integer ib,ie,jb,je,id,jd
      ppspec=pp(ipref,jpref)
      do 11 i=ib,ie
        do 21 j=jb,je
          pp(i,j)=pp(i,j)-ppspec
21      continue
11    continue
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine pcor(p,pp,ib,ie,jb,je,id,jd)
c
      real pp(42,302), p(42,302)
      integer ib,ie,jb,je,id,jd
      do 11 i=ib,ie
        do 21 j=jb,je
          p(i,j)=p(i,j) + pp(i,j)
21      continue
11    continue
      return
      end
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine velcor(u,v,depp,dnpp,pp,ib,ie,jb,je,id,jd)
c
      real pp(42,302), u(42,302), v(42,302), depp(42,302), dnpp(42,302)
      integer ib,ie,jb,je,id,jd
      do 11 i=ib,ie-1
        do 21 j=jb,je
          u(i,j)=u(i,j) + depp(i,j)*(pp(i,j)-pp(i+1,j))
21      continue
11    continue
      do 31 j=jb,je-1
        do 41 i=ib,ie
          v(i,j)=v(i,j) + dnpp(i,j)*(pp(i,j)-pp(i,j+1))
41      continue
```

```
31    continue
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
c  Use to apply block correction to (left/right) rows of control
c  volumes in a rectangular grid.  All boundary conditions must be
c  absorbed into the interior volumes and all edge coefficients
c  (i.e., an(i,je), aw(ib,j), as(i,jb) and ae(ie,j)) must be zero.
c
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine bcorj(phi,ap,ae,aw,an,as,bp,
     &                   ib,ie,jb,je,id,jd)
c
      real phi(42,302), ap(42,302), bp(42,302)
      real ae(42,302), aw(42,302), an(42,302), as(42,302)
      real ahatp(302), ahatn(302), ahats(302), bhatp(302)
      real addcor(302)
      do 10 j=jb,je
          addcor(j)=0.
          ahatp(j)=0.
          ahatn(j)=0.
          ahats(j)=0.
          bhatp(j)=0.
        do 20 i=ib,ie
            aeil=ae(i-1,j)
            if(i.eq.ib) aeil=0.
            awil=aw(i+1,j)
            if(i.eq.ie) awil=0.
          ahatp(j)=ahatp(j)+ap(i,j)-aeil-awil
          ahatn(j)=ahatn(j)+an(i,j)
          ahats(j)=ahats(j)+as(i,j)
          bhatp(j)=bhatp(j)+bp(i,j)  + aeil*phi(i,j)
     &                               + awil*phi(i,j)
     &                               + as(i,j)*phi(i,j-1)
     &                               + an(i,j)*phi(i,j+1)
     &                               - ap(i,j)*phi(i,j)
20      continue
10    continue
c
      call tdma(addcor,ahatp,ahatn,ahats,bhatp,
     &          jb,je,jd)
c
      do 50 j=jb,je
        do 60 i=ib,ie
          phi(i,j)=phi(i,j) + addcor(j)
60      continue
50    continue
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
c  Use to apply block correction to (up/down) columns of control
```

```
c  volumes in a rectangular grid.  All boundary conditions must be
c  absorbed into the interior volumes and all edge coefficients
c  (i.e., an(i,je), aw(ib,j), as(i,jb) and ae(ie,j)) must be zero.
c
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine bcori(phi,ap,ae,aw,an,as,bp,
     &                    ib,ie,jb,je,id,jd)
c
      real phi(42,302), ap(42,302), bp(42,302)
      real ae(42,302), aw(42,302), an(42,302), as(42,302)
      real ahatp(302), ahate(302), ahatw(302), bhatp(302)
      real addcor(302)
      do 10 i=ib,ie
          addcor(i)=0.
          ahatp(i)=0.
          ahate(i)=0.
          ahatw(i)=0.
          bhatp(i)=0.
        do 20 j=jb,je
            anj1=an(i,j-1)
            if(j.eq.jb) anj1=0.
            asj1=as(i,j+1)
            if(j.eq.je) asj1=0.
          ahatp(i)=ahatp(i)+ap(i,j)-anj1-asj1
          ahate(i)=ahate(i)+ae(i,j)
          ahatw(i)=ahatw(i)+aw(i,j)
          bhatp(i)=bhatp(i)+bp(i,j)  + anj1*phi(i,j)
     &                               + asj1*phi(i,j)
     &                               + aw(i,j)*phi(i-1,j)
     &                               + ae(i,j)*phi(i+1,j)
     &                               - ap(i,j)*phi(i,j)
20      continue
10    continue
c
      call tdma(addcor,ahatp,ahate,ahatw,bhatp,
     &         ib,ie,id)
c
      do 50 j=jb,je
        do 60 i=ib,ie
          phi(i,j)=phi(i,j) + addcor(i)
60      continue
50    continue
      return
      end
C*************************************************************************
C                                                                        *
C  SUBROUTINE TO CALCULATE RAYLEIGH NUMBER OF GAS SPACE AS A FUNCTION    *
C  OF MEAN GAS TEMPERATURE (K), PRESSURE (KPA), GAS LAYER THICKNESS      *
C  (M) AND TEMPERATURE DIFFERENCE ACROSS THE GAS LAYER (K)               *
C                                                                        *
C  OTHER VALUES RETURNED INCLUDE:                                        *
C      THERMAL CONDUCTIVITY          K         [W/mK]                    *
C      SPECIFIC HEAT                 CP        [J/KgK]                   *
```

```
C       VISCOSITY                    VISC      [Kg/m sec]              *
C       THERMAL EXPANSION COEF.      BETA      [/K]                    *
C       DENSITY                      RHO       [Kg/m3]                 *
C                                                                      *
C   PARAMETERS SUPPLIED TO THIS SUBROUTINE INCLUDE:                    *
C       MEAN TEMPERATURE             TM        [K]                     *
C       PRESSURE                     P         [KPa]                   *
C       GAS LAYER THICKNESS          TASI      [m]                     *
C       TEMPERATURE DIFFERENCE       DT        [K]                     *
C       GAS IDENTIFIER               IDGAS     []                      *
C          IDGAS=1 for air, 2 for argon, 3 for krypton, 4 for argon/   *
C                 freon, 5 for argon/SF6, 6 for argon/krypton          *
C       GAS COMPONENT MOLE FRACTION MFRAC      []                      *
C                                                                      *
C***********************************************************************
      SUBROUTINE FRA(RA,TM,P,TASI,DT,IDGAS,MFRAC,K,CP,VISC,BETA,RHO)
      REAL K,MHAT,MFRAC(2),KP(2),KDP(2),M(2),CHI(2,2),XSI(2,2)
      G=9.8064
C      g=9.8064 at Waterloo
      Z=FZ(TM,IDGAS)
      K=FK(TM,IDGAS,MFRAC)
      CPHAT=FCP(TM,IDGAS,MHAT,MFRAC)
      VISC=FVISC(TM,IDGAS,MFRAC)
      R=8.3143
      DV=1.E6*P/R/R/TM*P/TM/TM/Z/Z*G*CPHAT*TASI*TASI*CPHAT*TASI*DT
      RA=DV*MHAT/K/CPHAT/VISC
      CP=1000. * CPHAT/MHAT
      VISC=VISC/1000.
      BETA=1./TM
      RHO=P*MHAT/(R*TM)
      RETURN
      END
C***********************************************************************
C                                                                      *
C   subroutine to CALCULATE THE THERMAL CONDUCTIVITY OF VARIOUS FILL   *
C   GASES (W/m K) AT PRESSURES LESS THAN ONE ATMOSPHERE AND AS A       *
C   function OF TEMPERATURE (K) IN THE RANGE OF 240K to 300K           *
C                                                                      *
C   IDGAS=1 FOR AIR, 2 FOR ARGON, 3 FOR KRYPTON,4 FOR ARGON/FREON,     *
C         5 FOR ARGON/SF6, 6 FOR ARGON/KRYPTON                         *
C                                                                      *
C***********************************************************************
      function FK(TM,IDGAS,MFRAC)
      integer TERM
      real MFRAC(2),K(2),KP(2),KDP(2),CHI(2,2),XSI(2,2),M(2),KPMIX
      real KDPMIX
      term=4
      goto (1,2,3,4,5,6), IDGAS
      write(TERM,22)
22    format(2X,'GAS IDENTIFIER ERROR IN S.R. FK.  AIR ASSUMED')
1     FK=(0.0953286+0.0033086*TM)*2.414E-2
      goto 9
2     FK=(0.151607+0.0031036*TM)*1.634E-2
      goto 9
```

```
3     FK=855.136E-6+28.6275E-6*TM
      go to 9
4     continue
      K(1) = (0.151607 + 0.0031036*TM)*1.634E-2
      KP(1) = 0.002212 + 5.1403E-5*TM
      KDP(1) = K(1) - KP(1)
      M(1) = 39.948
      K(2) = -0.0023603 + 3.9526E-5*TM
      KP(2) = 0.0004665 + 8.7428E-6*TM
      KDP(2) = K(2) - KP(2)
      M(2) = 120.91
      go to 66
5     continue
      K(1) = (0.151607 + 0.0031036*TM)*1.634E-2
      KP(1) = 0.002212 + 5.1403E-5*TM
      KDP(1) = K(1) - KP(1)
      M(1) = 39.948
      K(2) = 0.013
      KP(2) = -2.837E-4 + 1.205E-5*TM
      KDP(2) = K(2) - KP(2)
      M(2) = 146.1
      goto 66
6     continue
      K(1) = (0.151607 + 0.0031036*TM)*1.634E-2
      KP(1) = 0.002212 + 5.1403E-5*TM
      KDP(1) = K(1) - KP(1)
      M(1) = 39.948
      K(2)=855.136E-6+28.6275E-6*TM
      DV= 0.00234 + 0.0000783*TM
      KP(2) = (15./4.)*(8.317/83.8)*DV
      KDP(2) = K(2) - KP(2)
      M(2) = 83.8
      goto 66
66    continue
      if(MFRAC(1).eq.0.0) go to 70
      if(MFRAC(2).eq.0.0) go to 80
      go to 19
70    continue
      KPMIX = KP(2)
      KDPMIX = KDP(2)
      go to 41
80    continue
      KPMIX = KP(1)
      KDPMIX = KDP(1)
      go to 41
19    continue
      KPMIX = 0.
      do 20 I=1,2
      SUM = 0.
      do 30 J=1,2
      if(I.eq.J) go to 30
      DV1 = (1.+ (KP(I)/KP(J))**0.5*(M(I)/M(J))**0.25)**2
      DV2 = 2.828*(1.+ (M(I)/M(J)))**0.5
      DV3 = 1.+ 2.41*((M(I)-M(J))*(M(I)-0.142*M(J))/(M(I)+M(J))**2)
      CHI(I,J) = DV1/DV2
      XSI(I,J) = CHI(I,J)*DV3
      SUM = SUM + XSI(I,J)*MFRAC(J)/MFRAC(I)
30    continue
      KPMIX = KPMIX + KP(I)/(1.+SUM)
20    continue
      KDPMIX = 0.
      do 40 I=1,2
      SUM = 0.
      do 50 J=1,2
      if(I.eq.J) go to 50
      SUM = SUM + CHI(I,J)*MFRAC(J)/MFRAC(I)
50    continue
      KDPMIX = KDPMIX + KDP(I)/(1.+SUM)
40    continue
41    continue
      FK = KPMIX + KDPMIX
      MHAT = 0.
      do 60 I=1,2
      MHAT = MHAT + M(I)*MFRAC(I)
60    continue
9     return
      end
C***********************************************************************
C                                                                      *
C  subroutine to CALCULATE THE SPECIFIC HEAT AT CONSTANT PRESSURE      *
C  (J/GMOLE K) OF VARIOUS FILL GASES AT PRESSURES LESS THAN ONE        *
C  ATMOSPHERE AND AS A function OF TEMPERATURE (K) OVER THE TEMP-      *
C  RANGE OF 240K to 300K.                                              *
C                                                                      *
C  IDGAS=1 FOR AIR, 2 FOR ARGON, 3 FOR KRYPTON, 4 FOR ARGON/FREON,     *
C        5 FOR ARGON/SF6, 6 FOR ARGON/KRYPTON                          *
C                                                                      *
C***********************************************************************
      function FCP(TM,IDGAS,MHAT,MFRAC)
      integer TERM
      real MFRAC(2),MHAT
      term=4
      goto (1,2,3,4,5,6),IDGAS
      write(TERM,44)
44    format(2X,'ERROR IN GAS IDENTIFIER AT S.R. FCP - AIR ASSUMED')
1     CP=(3.4898964+0.0000511*TM)*0.287041
      MHAT=28.97
      FCP = CP*MHAT
      goto 9
2     CP=(2.525325-0.0000661*TM)*0.208152
      MHAT=39.948
      FCP = CP*MHAT
      goto 9
3     CP=0.2497
      MHAT=83.8
      FCP = CP*MHAT
      go to 9
4     continue
```

```
      CP1 = (2.525325 - 0.0000661*TM)*0.208152
      CP2 =  0.249 + 0.001169*TM
      MHAT = MFRAC(1)*39.948 + MFRAC(2)*120.91
      FCP = MHAT*(MFRAC(1)*CP1 + MFRAC(2)*CP2)
      go to 9
5     continue
      CP1 = (2.5253 - 0.0000661*TM)*0.20815
      CP2 = 0.4186
      MHAT = MFRAC(1)*39.948 + MFRAC(2)*146.1
      FCP = MHAT*(MFRAC(1)*CP1 + MFRAC(2)*CP2)
      goto 9
6     continue
      CP1 = (2.5253 - 0.0000661*TM)*0.20815
      CP2 = 0.2497
      MHAT = MFRAC(1)*39.948 + MFRAC(2)*83.8
      FCP = MHAT*(MFRAC(1)*CP1 + MFRAC(2)*CP2)
      goto 9
9     return
      end
C***********************************************************************
C                                                                      *
C  subroutine to CALCULATE THE VISCOSITY OF VARIOUS FILL GASES         *
C  AT PRESSURES LESS THAN ONE ATMOSPHERE AND AS A function OF          *
C  TEMPERATURE IN THE RANGE OF 240K to 300K      [g/m sec]             *
C                                                                      *
C  IDGAS=1 FOR AIR, 2 FOR ARGON, 3 FOR KRYPTON, 4 FOR ARGON/FREON,     *
C       5 FOR ARGON/SF6, 6 FOR ARGON/KRYPTON                           *
C                                                                      *
C***********************************************************************
      function FVISC(TM,IDGAS,MFRAC)
      integer TERM
      real MFRAC(2),MHAT1,MHAT2
      term=4
      goto (1,2,3,4,5,6),IDGAS
      write(TERM,33)
33    format(2X,'GAS IDENTIFIER ERROR IN S.R. FPR - AIR ASSUMED')
1     FVISC=0.0035165 + 0.0000498*TM
      goto 9
2     FVISC=0.003618 + 0.0000644*TM
      goto 9
3     FVISC=0.00234 + 0.0000783*TM
      go to 9
4     VISC1 = 0.003618 + 0.0000644*TM
      VISC2 =-0.007614 + 7.428E-5*TM
      MHAT1 = 39.948
      MHAT2 = 120.91
      go to 66
5     continue
      VISC1 = 0.003618 + 0.0000644*TM
      VISC2 = 7.214E-4 + 4.928E-5*TM
      MHAT1 = 39.948
      MHAT2 = 146.1
      goto 66
6     continue
```

```
      VISC1 = 0.003618 + 0.0000644*TM
      VISC2 = 0.00234 + 0.0000783*TM
      MHAT1 = 39.948
      MHAT2 = 83.8
      goto 66
 66   continue
      if(MFRAC(1).eq.0.0) go to 40
      if(MFRAC(2).eq.0.0) go to 50
      go to 10
 40   FVISC = VISC2
      go to 9
 50   FVISC = VISC1
      go to 9
 10   continue
      DV1 = (1.+ (VISC1/VISC2)**0.5*(MHAT2/MHAT1)**0.25)**2
      DV2 = 2.8284*(1.+ MHAT1/MHAT2)**0.5
      CHI12 = DV1/DV2
      CHI21 = CHI12*(VISC2/VISC1)*(MHAT1/MHAT2)
      DV3 = VISC1/(1.+ CHI12*MFRAC(2)/MFRAC(1))
      DV4 = VISC2/(1.+ CHI21*MFRAC(1)/MFRAC(2))
      FVISC = DV3+DV4
 9    return
      end
C***********************************************************************
C                                                                      *
C  SUBROUTINE TO CALCULATE THE COMPRESSIBILITY FACTOR OF VARIOUS       *
C  FILL GASES FOR PRESSURES LESS THAN ONE ATMOSPHERE AND AS A FUN-     *
C  CTION OF TEMPERATURE (K) OVER THE TEMPERATURE RANGE OF 240K TO      *
C  300K                                                                *
C                                                                      *
C  IDGAS=1 FOR AIR, 2 FOR ARGON, 3 FOR KRYPTON, 4 FOR ARGON/FREON      *
C        5 FOR ARGON/SF6                                               *
C                                                                      *
C***********************************************************************
      FUNCTION FZ(TM, IDGAS)
      FZ=1.
      RETURN
      END
```

```
c
c  igurad.for
c
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
c  This subroutine sets up the A matrix, Amat, in the equation
c  A.X=B.  The vector X contains the irradiance values for the
c  solution of the radiant exchange problem in a rectangular
c  cavity with nsurf surfaces.  Amat is calculated given the
c  various geometric exchange factors, e(from,to), and the
c  diffuse reflectance values of the west wall, rwwd, east
c  wall, rewd, and the seals, rseal.
c
      subroutine setupa(amat,e,rwwd,rewd,rseal,
     &                  matdim,tl,tr,br,nsurf)
      real               amat(200,200), term
      real e(200,200)
      integer row, column
      integer tl, tr, br
      integer from, to
      do 10 row=1,matdim
        do 20 column=1,matdim
          amat(row,column)=0.
20      continue
10    continue
      do 30 from=1,nsurf
        amat(from,from)=1.
        do 40 to=1,tl-1
c         term=dble(e(from,to)*rwwd)
          term=    (e(from,to)*rwwd)
          amat(from,to)=amat(from,to)-term
40      continue
        do 50 to=tl,tr
c         term=dble(e(from,to)*rseal)
          term=    (e(from,to)*rseal)
          amat(from,to)=amat(from,to)-term
50      continue
        do 60 to=tr+1,br-1
c         term=dble(e(from,to)*rewd)
          term=    (e(from,to)*rewd)
          amat(from,to)=amat(from,to)-term
60      continue
        do 70 to=br,nsurf
c         term=dble(e(from,to)*rseal)
          term=    (e(from,to)*rseal)
          amat(from,to)=amat(from,to)-term
70      continue
30    continue
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
c  Given the x/y indices defining the cavity, ibc, iec, jbc,
c  jec and defining the centre-glass portion of the cavity,
c  jbcg and jecg, the set of specular exchange factors
c  e(from,to) are calculated for all the nsurf surfaces of
c  the cavity accounting for specular reflection at the west
c  and east walls only (specified by rwws and rews, respectively).
c
c  Also returned are the indices tl and tr for the left and right
c  extremes of the top surface, respectively;  br and nsurf for the
c  right and left extremes of the bottom surface, respectively; and
c  lcg and rcg for the left and right centre-glass portions of the
c  vertical walls, respectively.
c  The radiative surface numbering system runs from 1 at the bottom
c  most segment on the west (or left) wall then clockwise to surface
c  number nsurf at the left most end of the bottom wall.
c
      Subroutine exfact(e,nsurf,tl,tr,br,lx,ly,rwws,rews,
     &                      lcg,rcg,
     &                      ibc,iec,jbc,jec,
     &                      jbcg,jecg,matdim,ierr,esumax,esumin,oput)
      integer from, to
      integer lt, tl, tr, rt, rb, br
      integer lcg, lcgm, lcgp
      integer rcg, rcgm, rcgp
      integer oput
      real e(200,200)
      real lx, ly
      do 10 from=1,matdim
        do 20 to=1,matdim
          e(from,to)=0.
20      continue
10    continue
c
      ierr=0
      nsurf = 2 + 4*(jbcg-jbc) + 2*(iec-ibc+1)
      if(nsurf.gt.matdim) then
         ierr=1
         goto 99
      endif
c
      ndivx=iec-ibc+1
      ndivy=jec-jbc+1
      lcgm=jbcg-jbc
      lcg=lcgm+1
      lcgp=lcg+1
      lt=lcg+(jec-jecg)
      tl=lt+1
      tr=lt+ndivx
      rt=tr+1
      rcg=rt+(jec-jecg)
      rcgm=rcg-1
      rcgp=rcg+1
      rb=rcg+(jbcg-jbc)
      br=rb+1
c
c       write(oput,*) jbcg, jecg
```

```
c         write(oput,*) tl,tr,br,nsurf
c         write(oput,*) lcg,rcg
c
       do 30 from=1,nsurf
           call surfxy(from,xlp,ylp,x2p,y2p,
     &                 lx,ly,ndivx,ndivy,
     &                 ibc,iec,jbc,jec,jbcg,jecg,
     &                 lcg,tl,tr,
     &                 rcg,br)
c
         do 40 to=from,nsurf
           call surfxy(to  ,xlnb,ylnb,x2nb,y2nb,
     &                 lx,ly,ndivx,ndivy,
     &                 ibc,iec,jbc,jec,jbcg,jecg,
     &                 lcg,tl,tr,
     &                 rcg,br)
           ato=fdist(xlnb,ylnb,x2nb,y2nb)
           afrom=fdist(xlp,ylp,x2p,y2p)
           nterm=100
           call ecalc(edv,afrom,ato,from,to,lx,ly,
     &                xlp,ylp,x2p,y2p,
     &                xlnb,ylnb,x2nb,y2nb,
     &                tl,tr,br,
     &                rwws,rews,nterm,oput)
           e(from,to) = edv
c
           if(to.ne.from) e(to,from)=e(from,to)*(afrom/ato)
40       continue
30     continue
       esumax=-1000.
       esumin= 1000.
       do 50 from=1,nsurf
         call chksum(e,from,sum,tl,tr,br,nsurf,rwws,rews)
         if(sum.gt.esumax) esumax=sum
         if(sum.lt.esumin) esumin=sum
50     continue
99     return
       end
c   /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
c   This subroutine classifies the location of various cavity
c   surfaces according to the numbering index of each surface.
c   Individual specular exchange factors, e(from,to), are then
c   calculated by calling specialized subroutines (e.g., the
c   subroutine elt(........) calculates exchange factors from
c   the left wall to the top wall, ell calculates exchange
c   factors from the left wall to the left wall;  you can
c   figure out the others.)
c
            subroutine ecalc(edv,afrom,ato,from,to,lx,ly,
     &                       xlp,ylp,x2p,y2p,
     &                       xlnb,ylnb,x2nb,y2nb,
     &                       tl,tr,br,
     &                       rwws,rews,nterm,oput)
```

```
      integer from,to,tl,tr,br
      integer oput
      real lx, ly
      edv=0.
c  Exchange factors between left wall
c  and left/top/right/bottom wall
      if(from.lt.tl) then
        if(to.lt.tl)
     &      call ell(edv,x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,
     &               afrom,ato,rwws,rews,nterm,lx,ly,oput)
        if((to.ge.tl).and.(to.le.tr))
     &      call elt(edv,x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,
     &               afrom,ato,rwws,rews,nterm,lx,ly,oput)
        if((to.gt.tr).and.(to.lt.br))
     &      call elr(edv,x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,
     &               afrom,ato,rwws,rews,nterm,lx,ly,oput)
        if(to.ge.br)
     &      call elt(edv,x1p,ly-y1p,x2p,ly-y2p,
     &               x1nb,ly-y1nb,x2nb,ly-y2nb,
     &               afrom,ato,rwws,rews,nterm,lx,ly,oput)
      goto 99
      endif

c  Exchange factors between top wall
c  and top/right/bottom wall
      if((from.ge.tl).and.(from.le.tr)) then
        if((to.ge.tl).and.(to.le.tr)) goto 99
        if((to.gt.tr).and.(to.lt.br))
     &      call etr(edv,x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,
     &               afrom,ato,rwws,rews,nterm,lx,ly,oput)
        if(to.ge.br)
     &      call etb(edv,x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,
     &               afrom,ato,rwws,rews,nterm,lx,ly,oput)
      goto 99
      endif

c  Exchange factors between east wall
c  and right/bottom wall
      if((from.gt.tr).and.(from.lt.br))then
        if((to.gt.tr).and.(to.lt.br))
     &      call ell(edv,lx-x1p,y1p,lx-x2p,y2p,
     &               lx-x1nb,y1nb,lx-x2nb,y2nb,
     &               afrom,ato,rews,rwws,nterm,lx,ly,oput)
        if(to.ge.br)
     &      call elt(edv,lx-x1p,ly-y1p,lx-x2p,ly-y2p,
     &               lx-x1nb,ly-y1nb,lx-x2nb,ly-y2nb,
     &               afrom,ato,rews,rwws,nterm,lx,ly,oput)
      goto 99
      endif

c  Exchange factors between bottom wall
c  and bottom wall

c  These are all equal to zero because the seals are assumed
```

```
c  to have zero specular reflectance

99    return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
c  Given the surface number, isurf, the x and y dimensions
c  at each end of the surface are calculated.
c
          subroutine surfxy(isurf,x1,y1,x2,y2,
     &                           lx,ly,ndivx,ndivy,
     &                           ibc,iec,jbc,jec,jbcg,jecg,
     &                           lcg,tl,tr,
     &                           rcg,br)
      integer lt, tl, tr, rt, rb, br
      integer lcg, lcgm, lcgp
      integer rcg, rcgm, rcgp
      real lx, ly
c  Figure out x coordinates
      x1=0.
      x2=0.
      if(isurf.lt.tl) goto 88
      x1=lx
      x2=lx
      if((isurf.gt.tr).and.(isurf.lt.br)) goto 88
      if(isurf.ge.br) then
        x1=lx - (float(isurf-br+1)/float(ndivx))*lx
        x2=lx - (float(isurf-br  )/float(ndivx))*lx
      goto 88
      endif
        x1=     (float(isurf-tl  )/float(ndivx))*lx
        x2=     (float(isurf-tl+1)/float(ndivx))*lx
88    continue
c  Figure out y coordinates
c  bottom
      y1=0.
      y2=0.
      if(isurf.ge.br) goto 99
c  top
      y1=ly
      y2=ly
      if((isurf.ge.tl).and.(isurf.le.tr)) goto 99
c  left wall
      if(isurf.lt.lcg) then
      y1 =      ly*float(isurf-1)/float(ndivy)
      y2 =      ly*float(isurf  )/float(ndivy)
      goto 99
      endif
      if(isurf.eq.lcg) then
      y1 =     ly*float(jbcg-jbc  )/float(ndivy)
      y2 =      ly*float(jecg-jbc+1)/float(ndivy)
      goto 99
      endif
      if((isurf.gt.lcg).and.(isurf.lt.tl))then
      y1 =ly - ly*float(tl-isurf  )/float(ndivy)
      y2 =ly - ly*float(tl-isurf-1)/float(ndivy)
      goto 99
      endif
c  right wall
      if((isurf.gt.tr).and.(isurf.lt.rcg))then
      y1 = ly - ly*float(isurf-tr)/float(ndivy)
      y2 = ly - ly*float(isurf-tr-1)/float(ndivy)
      goto 99
      endif
      if(isurf.eq.rcg) then
      y1 =      ly*float(jbcg-jbc  )/float(ndivy)
      y2 =       ly*float(jecg-jbc+1)/float(ndivy)
      goto 99
      endif
      if((isurf.gt.rcg).and.(isurf.lt.br))then
      y1 =      ly*float(br-isurf-1)/float(ndivy)
      y2 =      ly*float(br-isurf  )/float(ndivy)
      endif
99    return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
      subroutine elr(edv,x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,
     &               afrom,ato,rwws,rews,nterm,lx,ly,oput)
      real lx, ly
      integer oput
      edv=fxstring(x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,afrom,ato,oput)
c  add specular reflectance terms
      do 10 i=2,nterm*2,2
        x1im=x1nb + float(i)*lx
        x2im=x2nb + float(i)*lx
        nrw=i/2
        nre=i/2
       if(nrw+nre.ne.i) write(oput,*) 'Error in s.r. elr'
        atten=1.
        atten=atten*rwws**nrw
        atten=atten*rews**nre
        term=atten*fxstring(x1p,y1p,x2p,y2p,x1im,y1nb,x2im,y2nb,
     &                      afrom,ato,oput)
        edv=edv+term
10    continue
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
      subroutine elt(edv,x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,
     &               afrom,ato,rwws,rews,nterm,lx,ly,oput)
      real lx, ly
      integer oput
      edv=fxstring(x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,afrom,ato,oput)
c  add specular reflectance terms
      do 10 i=2,nterm*2,2
        x1im=x1nb + float(i)*lx
        x2im=x2nb + float(i)*lx
        nrw=i/2
        nre=i/2
      if(nrw+nre.ne.i) write(oput,*) 'Error in s.r. elt'
        atten=1.
        atten=atten*rwws**nrw
        atten=atten*rews**nre
        term=atten*fxstring(x1p,y1p,x2p,y2p,x1im,y1nb,x2im,y2nb,
     &                      afrom,ato,oput)
        edv=edv+term
10    continue
      do 20 i=1,nterm*2-1,2
        x1im=float(i+1)*lx - x1nb
        x2im=float(i+1)*lx - x2nb
        nrw=(i+1)/2 -1
        nre=(i+1)/2
      if(nrw+nre.ne.i) write(oput,*) 'Error in s.r. elt'
        atten=1.
        if(nrw.ne.0) atten=atten*rwws**nrw
        if(nre.ne.0) atten=atten*rews**nre
        term=atten*fxstring(x1p,y1p,x2p,y2p,x1im,y1nb,x2im,y2nb,
     &                      afrom,ato,oput)
        edv=edv+term
20    continue
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
      subroutine etb(edv,x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,
     &               afrom,ato,rwws,rews,nterm,lx,ly,oput)
      real lx, ly
      integer oput
      edv=fxstring(x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,afrom,ato,oput)
c  add specular reflectance terms
      do 10 i=2,nterm*2,2
        x1im=x1nb - float(i)*lx
        x2im=x2nb - float(i)*lx
        nrw=i/2
        nre=i/2
      if(nrw+nre.ne.i) write(oput,*) 'Error in s.r. etb'
        atten=1.
        atten=atten*rwws**nrw
        atten=atten*rews**nre
        term=atten*fxstring(x1p,y1p,x2p,y2p,x1im,y1nb,x2im,y2nb,
     &                      afrom,ato,oput)
        edv=edv+term
        x1im=x1nb + float(i)*lx
        x2im=x2nb + float(i)*lx
        term=atten*fxstring(x1p,y1p,x2p,y2p,x1im,y1nb,x2im,y2nb,
     &                      afrom,ato,oput)
        edv=edv+term
10    continue
      do 20 i=1,nterm*2-1,2
        x1im=-lx*float(i-1) - x1nb
        x2im=-lx*float(i-1) - x2nb
        nrw=(i+1)/2
        nre=(i+1)/2 - 1
      if(nrw+nre.ne.i) write(oput,*) 'Error in s.r. etb'
        atten=1.
        if(nrw.ne.0) atten=atten*rwws**nrw
        if(nre.ne.0) atten=atten*rews**nre
        term=atten*fxstring(x1p,y1p,x2p,y2p,x1im,y1nb,x2im,y2nb,
     &                      afrom,ato,oput)
        edv=edv+term
        x1im= lx*float(i+1) - x1nb
        x2im= lx*float(i+1) - x2nb
        nrw=(i+1)/2 - 1
        nre=(i+1)/2
      if(nrw+nre.ne.i) write(oput,*) 'Error in s.r. etb'
        atten=1.
        if(nrw.ne.0) atten=atten*rwws**nrw
        if(nre.ne.0) atten=atten*rews**nre
        term=atten*fxstring(x1p,y1p,x2p,y2p,x1im,y1nb,x2im,y2nb,
     &                      afrom,ato,oput)
        edv=edv+term
20    continue
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
      subroutine etr(edv,x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,
     &               afrom,ato,rwws,rews,nterm,lx,ly,oput)
      real lx, ly
      integer oput
      edv=fxstring(x1p,y1p,x2p,y2p,x1nb,y1nb,x2nb,y2nb,afrom,ato,oput)
c  add specular reflectance terms
      do 10 i=2,nterm*2,2
        x1im=x1nb + float(i)*lx
        x2im=x2nb + float(i)*lx
        nrw=i/2
        nre=i/2
      if(nrw+nre.ne.i) write(oput,*) 'Error in s.r. etr'
        atten=1.
        atten=atten*rwws**nrw
        atten=atten*rews**nre
        term=atten*fxstring(x1p,y1p,x2p,y2p,x1im,y1nb,x2im,y2nb,
     &                      afrom,ato,oput)
        edv=edv+term
10    continue
      do 20 i=1,nterm*2-1,2
        x1im=-lx*float(i-1) - x1nb
        x2im=-lx*float(i-1) - x2nb
        nrw=(i+1)/2
        nre=(i+1)/2 - 1
      if(nrw+nre.ne.i) write(oput,*) 'Error in s.r. etr'
        atten=1.
        if(nrw.ne.0) atten=atten*rwws**nrw
        if(nre.ne.0) atten=atten*rews**nre
        term=atten*fxstring(xlp,ylp,x2p,y2p,xlim,ylnb,x2im,y2nb,
     &                      afrom,ato,oput)
        edv=edv+term
20    continue
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
      subroutine ell(edv,xlp,ylp,x2p,y2p,xlnb,ylnb,x2nb,y2nb,
     &               afrom,ato,rwws,rews,nterm,lx,ly,oput)
      real lx, ly
      integer oput
      edv=0.
c  add specular reflectance terms
      do 20 i=1,nterm*2-1,2
c       xlim=float(i+1)*lx - xlnb
c       x2im=float(i+1)*lx - x2nb
c  xlnb and x2nb should be equal to zero
        xlim=float(i+1)*lx
        x2im=float(i+1)*lx
        nrw=(i+1)/2 -1
        nre=(i+1)/2
      if(nrw+nre.ne.i) write(oput,*) 'Error in s.r. ell'
        atten=1.
        if(nrw.ne.0) atten=atten*rwws**nrw
        if(nre.ne.0) atten=atten*rews**nre
        term=atten*fxstring(xlp,ylp,x2p,y2p,xlim,ylnb,x2im,y2nb,
     &                      afrom,ato,oput)
        edv=edv+term
20    continue
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
c  Hottels crossed-string method is used to calculate the shape
c  factor from surface p to surface nb assuming neither surface
c  can see the back surface of the other.
c
      function fxstring(xlp,ylp,x2p,y2p,xlnb,ylnb,x2nb,y2nb,afrom,ato,
     &                  oput)
      integer oput
      side1=fdist(xlp,ylp,xlnb,ylnb)
      side2=fdist(x2p,y2p,x2nb,y2nb)
      cross1=fdist(xlp,ylp,x2nb,y2nb)
      cross2=fdist(x2p,y2p,xlnb,ylnb)
      dv = (cross1+cross2-side1-side2)/(2.*afrom)
      if(dv.lt.0.) dv=-1.*dv
      if(dv.gt.1.) write(oput,*) 'Error in s.r. fxstring'
      fxstring=dv
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
c  Function subroutine that calculates the distance between
```

```
c  two points in a 2-D plane.
      function fdist(x1,y1,x2,y2)
      fdist=(((x2-x1)**2) + ((y2-y1)**2))**0.5
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
c  Calculates a check-sum.  The sum of the specular exchange
c  factors from surface from, to all the surfaces in the
c  cavity (multiplied by (1-specular reflectance of the other
c  surface in each case) should be equal to 1.  In other words
c  all of the radiation leaving a surface must end up somewhere
c  in the enclosure.
c
      subroutine chksum(e,from,sum,tl,tr,br,nsurf,rwws,rews)
      real e(200,200)
      integer from, to, tl,tr,br
      sum=0.
      do 10 to=1,nsurf
        refs=0.
        if(to.lt.tl) refs=rwws
        if((to.gt.tr).and.(to.lt.br)) refs=rews
        sum=sum+e(from,to) * (1.-refs)
10    continue
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
c  Given the temperature of the all the control volumes in the
c  problem domain, T, the surface temperatures of the surfaces
c  facing the cavity are calculated.
c
      subroutine setemp(tsurf,t,nsurf,con,kg,kseal,
     &                  diep,djnp,tg,dyseal,
     &                  ibc,iec,jbc,jec,jbcg,jecg,oput)
      real kg, kseal
      real tsurf(200), t(42,302)
      integer oput
      isurf=0
c  left wall, bottom to top ------------------------------------
      denom =  (kg/tg) + (con/diep)
      do 10 j=jbc,jbcg-1
        isurf=isurf+1
        tsurf(isurf)=(t(ibc-1,j)*(kg/tg)+t(ibc,j)*(con/diep))/denom
10    continue
c
c  left centre-glass
      isurf=isurf+1
      sum=0.
      do 20 j=jbcg,jecg
        sum=sum + (t(ibc-1,j)*(kg/tg)+t(ibc,j)*(con/diep))/denom
20    continue
      tsurf(isurf) = sum/float(jecg-jbcg+1)
c
```

```
      do 30 j=jecg+1,jec
        isurf=isurf+1
        tsurf(isurf)=(t(ibc-1,j)*(kg/tg)+t(ibc,j)*(con/diep))/denom
30    continue
c
c   top wall, left to right -------------------------------------
      denom =  (kseal/dyseal) + (con/djnp)
      do 40 i=ibc,iec
        isurf=isurf+1
        tsurf(isurf)=(t(i,jec+1)*(kseal/dyseal) +
     &                                 t(i,jec)*(con/djnp))/denom
40    continue
c
c  right wall, top to bottom  ----------------------------------
      denom =  (kg/tg) + (con/diep)
      do 50 jdv=1,jec-jecg
        j=jec+1-jdv
        isurf=isurf+1
        tsurf(isurf)=(t(iec+1,j)*(kg/tg)+t(iec,j)*(con/diep))/denom
50    continue
c
c right centre-glass
      isurf=isurf+1
      sum=0.
      do 60 j=jbcg,jecg
        sum=sum + (t(iec+1,j)*(kg/tg)+t(iec,j)*(con/diep))/denom
60    continue
      tsurf(isurf) = sum/float(jecg-jbcg+1)
c
      do 70 jdv=1,jbcg-jbc
        j=jbcg-jdv
        isurf=isurf+1
        tsurf(isurf)=(t(iec+1,j)*(kg/tg)+t(iec,j)*(con/diep))/denom
70    continue
c
c   bottom wall, right to left  ---------------------------------
      denom =  (kseal/dyseal) + (con/djnp)
      do 80 idv=1,iec-ibc+1
        i=iec+1-idv
        isurf=isurf+1
        tsurf(isurf)=(t(i,jbc-1)*(kseal/dyseal) +
     &                                 t(i,jbc)*(con/djnp))/denom
80    continue
      if(isurf.ne.nsurf) write(oput,*) 'Error in s.r. setemp',
     &                                     isurf,nsurf
c
      do 90 isurf=1,nsurf
        tsurf(isurf) = tsurf(isurf) + 273.159
90    continue
c
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
```

```
      subroutine setrhs(rhsvec,e,tsurf,rews,rwws,rewd,rwwd,rseal,
     &                  tl,tr,br,nsurf)
      integer from, to
      integer tl, tr, br
      real rhsvec(200), e(200,200), tsurf(200)
      sig=5.6693
      do 100 from=1,nsurf
      sum=0.
c  left wall
        epsi = 1.-rwws-rwwd
        do 10 to=1,tl-1
          sum = sum + e(from,to)*epsi*sig*(tsurf(to)/100.)**4
10      continue
c   top wall
        epsi = 1.- rseal
        do 20 to=tl,tr
          sum = sum + e(from,to)*epsi*sig*(tsurf(to)/100.)**4
20      continue
c  right wall
        epsi = 1.-rews-rewd
        do 30 to=tr+1,br-1
          sum = sum + e(from,to)*epsi*sig*(tsurf(to)/100.)**4
30      continue
c  bottom wall
        epsi = 1.-rseal
        do 40 to=br,nsurf
          sum = sum + e(from,to)*epsi*sig*(tsurf(to)/100.)**4
40      continue
        rhsvec(from)=sum
100   continue
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
c  This subroutine from Numerical Recipes by Press, Flannery,
c  Teukolsky and Vetterling - see Chapter 2. (LU Decomposition)
c
c  Given an NxN matrix A, with physical dimension NPxNP, this
c  this subroutine replaces it by the LU decomposition of a
c  rowwise permutation of itself.  A and N are input.  A is
c  output;  INDX is an output vector which records the row
c  permutation effected by the partial pivoting;  D is output
c  as +/- 1 depending on whether the number of row interchanges
c  was even or odd, respectively.  This routine is used in
c  combination with LUBKSB to solve linear equations or to
c  invert a matrix.
c
      SUBROUTINE LUDCMP(A,N,NP,INDX,D)
      PARAMETER (TINY=1.0E-20)
c     DIMENSION A(NP,NP),INDX(NP),VV(NP)
      real a(200,200), vv(200)
      integer indx(200)
      D=1.
      DO 12 I=1,N
```

```
        AAMAX=0.
        DO 11 J=1,N
          IF (ABS(A(I,J)).GT.AAMAX) AAMAX=ABS(A(I,J))
11      CONTINUE
        IF (AAMAX.EQ.0.) PAUSE 'Singular matrix.'
        VV(I)=1./AAMAX
12    CONTINUE
      DO 19 J=1,N
        IF (J.GT.1) THEN
          DO 14 I=1,J-1
            SUM=A(I,J)
            IF (I.GT.1)THEN
              DO 13 K=1,I-1
                SUM=SUM-A(I,K)*A(K,J)
13            CONTINUE
              A(I,J)=SUM
            ENDIF
14        CONTINUE
        ENDIF
        AAMAX=0.
        DO 16 I=J,N
          SUM=A(I,J)
          IF (J.GT.1)THEN
            DO 15 K=1,J-1
              SUM=SUM-A(I,K)*A(K,J)
15          CONTINUE
            A(I,J)=SUM
          ENDIF
          DUM=VV(I)*ABS(SUM)
          IF (DUM.GE.AAMAX) THEN
            IMAX=I
            AAMAX=DUM
          ENDIF
16      CONTINUE
        IF (J.NE.IMAX)THEN
          DO 17 K=1,N
            DUM=A(IMAX,K)
            A(IMAX,K)=A(J,K)
            A(J,K)=DUM
17        CONTINUE
          D=-D
          VV(IMAX)=VV(J)
        ENDIF
        INDX(J)=IMAX
        IF(J.NE.N)THEN
          IF(A(J,J).EQ.0.)A(J,J)=TINY
          DUM=1./A(J,J)
          DO 18 I=J+1,N
            A(I,J)=A(I,J)*DUM
18        CONTINUE
        ENDIF
19    CONTINUE
      IF(A(N,N).EQ.0.)A(N,N)=TINY
      RETURN
      END
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
c  This subroutine from Numerical Recipes by Press, Flannery,
c  Teukolsky and Vetterling - see Chapter 2. (LU Decomposition)
c
c  Solves the set of N linear equations, A.X=B.  Here, A is input
c  not as the matrix A but rather as its LU decomposition,
c  determined by the subroutine LUDCMP.  INDX is returned as the
c  permutation vector returned by LUDCMP.  B is input as the RHS
c  vector B,  and returns with the solution vector, X.  A, N, NP
c  and INDX are not modified by this routine and can be left in
c  place for successive calls with different RHS's B.  This routine
c  takes into account the possibility that B will begin with many
c  zero elements, so it is efficient for use in matrix inversion.
c
      SUBROUTINE LUBKSB(A,N,NP,INDX,B)
c     DIMENSION A(NP,NP),INDX(NP),B(NP)
      real a(200,200), b(200)
      integer indx(200)
      II=0
      DO 12 I=1,N
        LL=INDX(I)
        SUM=B(LL)
        B(LL)=B(I)
        IF (II.NE.0)THEN
          DO 11 J=II,I-1
            SUM=SUM-A(I,J)*B(J)
11        CONTINUE
        ELSE IF (SUM.NE.0.) THEN
          II=I
        ENDIF
        B(I)=SUM
12    CONTINUE
      DO 14 I=N,1,-1
        SUM=B(I)
        IF(I.LT.N)THEN
          DO 13 J=I+1,N
            SUM=SUM-A(I,J)*B(J)
13        CONTINUE
        ENDIF
        B(I)=SUM/A(I,I)
14    CONTINUE
      RETURN
      END
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
c  This subroutine from Numerical Recipes by Press, Flannery,
c  Teukolsky and Vetterling - see Chapter 2. (LU Decomposition)
c
c  Improves a solution vector X, of the linear set of equations
c  A.X=B.  The matrix A and the vectors B and X are input, as is
c  the dimension N.  Also input is ALUD, the LU decomposition of
c  A as returned by LUDCMP and the vector INDX also returned by
```

```
c  that routine.  On output, only X is modified, to an improved
c  set of values.
c
      SUBROUTINE MPROVE(A,ALUD,N,NP,INDX,B,X)
c     PARAMETER (NMAX=100)
c     DIMENSION A(NP,NP),ALUD(NP,NP),INDX(NP),B(NP),X(NP),R(NP)
      real a(200,200), alud(200,200), b(200), x(200), r(200)
      integer indx(200)
      REAL*8 SDP
      DO 12 I=1,N
        SDP=-B(I)
        DO 11 J=1,N
          SDP=SDP+DBLE(A(I,J))*DBLE(X(J))
11      CONTINUE
        R(I)=SDP
12    CONTINUE
      CALL LUBKSB(ALUD,N,NP,INDX,R)
      DO 13 I=1,N
        X(I)=X(I)-R(I)
13    CONTINUE
      RETURN
      END
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
      subroutine calcsp(sp,h,tsurf,rews,rwws,rewd,rwwd,rseal,
     &                  tl,tr,br,nsurf)
      integer surf
      integer tl, tr, br
      real sp(200), h(200), tsurf(200)
      sig=5.6693
c  left wall
      epsi = 1.-rwws-rwwd
      do 10 surf=1,tl-1
        sp(surf)=epsi*(h(surf) - sig*(tsurf(surf)/100.)**4)
10    continue
c   top wall
      epsi = 1.- rseal
      do 20 surf=tl,tr
        sp(surf)=epsi*(h(surf) - sig*(tsurf(surf)/100.)**4)
20    continue
c  right wall
      epsi = 1.-rews-rewd
      do 30 surf=tr+1,br-1
        sp(surf)=epsi*(h(surf) - sig*(tsurf(surf)/100.)**4)
30    continue
c  bottom wall
      epsi = 1.-rseal
      do 40 surf=br,nsurf
        sp(surf)=epsi*(h(surf) - sig*(tsurf(surf)/100.)**4)
40    continue
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
```

```
      subroutine calcres(hresid,amat,xvec,bvec,nsurf)
      real amat(200,200), xvec(200), bvec(200)
      real hresid(200)
      integer column, row
      do 10 row=1,nsurf
        sum=0.
        do 20 column=1,nsurf
          sum = sum + amat(row,column) * xvec(column)
20      continue
        hresid(row)=sum-bvec(row)
10    continue
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
      subroutine checksp(spsum,sp,lx,ly,
     &                   ibc,iec,jbc,jec,jbcg,jecg,
     &                   tl,tr,br,lcg,rcg,nsurf)
      real sp(200), lx, ly
      integer tl, tr, br
      integer lcg, rcg
      ncent=jecg-jbcg+1
      sum=0.
      do 10 i=1,lcg-1
        sum=sum+sp(i)
10    continue
      sum=sum+float(ncent)*sp(lcg)
      do 20 i=lcg+1,tl-1
        sum=sum+sp(i)
20    continue
      spsum=ly*sum/float(jec-jbc+1)
c
      sum=0.
      do 30 i=tl,tr
        sum=sum+sp(i)
30    continue
      spsum=spsum + lx*sum/float(iec-ibc+1)
c
      sum=0.
      do 40 i=tr+1,rcg-1
        sum=sum+sp(i)
40    continue
      sum=sum+float(ncent)*sp(rcg)
      do 50 i=rcg+1,br-1
        sum=sum+sp(i)
50    continue
      spsum=spsum + ly*sum/float(jec-jbc+1)
c
      sum=0.
      do 60 i=br,nsurf
        sum=sum+sp(i)
60    continue
      spsum=spsum + lx*sum/float(iec-ibc+1)
c
```

```
      return
      end
c  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
c
      subroutine convsp(sp,rt,qt,tg,dyseal,cp,diep,djnp,
     &                   ib,ie,jb,je,
     &                   ibc,iec,jbc,jec,jbcg,jecg,
     &                   tl,tr,br,lcg,rcg,nsurf,oput)
      real rt(42,302), qt(42,302)
      real sp(200), lx, ly
      integer tl, tr, br
      integer lcg, rcg
      integer oput
      ncent=jecg-jbcg+1
c
      do 110 i=ib,ie
        do 120 j=jb,je
          qt(i,j)=0.
          rt(i,j)=0.
120     continue
110   continue
c
      isurf=0
c
c  left wall
c
      i=ibc-1
      do 10 j=jbc,jbcg-1
        isurf=isurf+1
        qt(i,j)=djnp*sp(isurf)/cp
10    continue
      isurf=isurf+1
      do 20 j=jbcg,jecg
        qt(i,j)=djnp*sp(isurf)/cp
20    continue
      do 30 j=jecg+1,jec
        isurf=isurf+1
        qt(i,j)=djnp*sp(isurf)/cp
30    continue
c
c  top wall
c
      j=jec+1
      do 40 i=ibc,iec
        isurf=isurf+1
        qt(i,j)=diep*sp(isurf)/cp
40    continue
c
c  right wall
c
      i=iec+1
      do 50 jdv=1,jec-jecg
        j=jec+1-jdv
        isurf=isurf+1
        qt(i,j)=djnp*sp(isurf)/cp
50    continue
      isurf=isurf+1
      do 60 jdv=1,jecg-jbcg+1
        j=jecg+1-jdv
        qt(i,j)=djnp*sp(isurf)/cp
60    continue
      do 70 jdv=1,jbcg-jbc
        j=jbcg-jdv
        isurf=isurf+1
        qt(i,j)=djnp*sp(isurf)/cp
70    continue
c
c  bottom wall
c
      j=jbc-1
      do 80 idv=1,iec-ibc+1
        i=iec+1-idv
        isurf=isurf+1
        qt(i,j)=diep*sp(isurf)/cp
80    continue
      if(isurf.ne.nsurf) write(oput,*) 'Error in s.r. convsp'
     &                   ,isurf,nsurf
      return
      end
```

```
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c  This routine calculates x and y coordinates for a set nodes
c  corresponding to the nodes at the centres of the finite volumes.
c  Nodes are also created along the walls, at the corners of the
c  cavity and at surfaces between differing materials.
c  Next, rectangular finite elements are defined with these nodes
c  located at the corners.
c  The node and element data are written to an output file in the
c  Neutral File format that can be used as input to the PATRAN
c  finite element graphics i/o package.
c  Node or element data can be suppressed using the two digits
c  of the integer flag, iopat, respectively.
c  Data are generated and written to the PATRAN neutral files for
c  the range of j from jb to jpate.
c  jpate must be less than jec (i.e., cavity cut off below upper
c  seal.
c
      subroutine nodel(lx,ly,ib,ie,jb,je,jpate,tww,twe,oput,
     &                     wseal,nseal,ibc,iec,jbc,jec,tn,tg,
     &                     title)
      real lx,ly
      integer oput
      integer config, cid, pspc(6)
      integer pid, ceid
      character*14 neutfn
      character*80 title
      character*12 date
      character*8 time
      character*12 version
      character*1 gtype
      if(jpate.gt.je) goto 99
      if(jpate.lt.jb) goto 99
c     title='VSML node & element output to PATRAN'
      date='17-May-90'
      time='16:06:34'
      version=' 2.3'
      iz=0
25    format(i2,8i8)
251   format(a80)
261   format(a12,a8,a12)
10    format(e16.9)
12    format(3e16.9)
13    format(i1,a1,3i8,2x,6i1)
22    format(4i8,3e16.9)
23    format(10i8)
      neutfn='nodeneut.pat'
      diep=lx/float(iec-ibc+1)
      djnp=ly/float(jec-jbc+1)
      dyseal=wseal/nseal
c     open(unit=oput,file=neutfn,status='unknown')
c  Title Card
      call head(it,id,iv,kc,n1,n2,n3,n4,n5)
      kc=1
      it=25
      write(oput,25) it,id,iv,kc,n1,n2,n3,n4,n5
      write(oput,251) title
c  Summary Data
      call head(it,id,iv,kc,n1,n2,n3,n4,n5)
      it=26
      kc=1
      nny=jpate-jbc+nseal+3
      nnx=ie-ib+11
      n1=nnx*nny
      n2=(nnx-1)*(nny-1)
      n3=1
      n4=4
      n5=1
      write(oput,25) it,id,iv,kc,n1,n2,n3,n4,n5
      write(oput,261) date, time, version
c  Node Data
      call head(it,id,iv,kc,n1,n2,n3,n4,n5)
      z=0.
      it=1
      iv=0
      kc=2
      icf=1
      gtype='T'
      ndf=0
      config=0
      cid=0
      do 300 idv=1,6
        pspc(idv)=0
300   continue
      do 100 jn=1,nny
        y=wseal+ly*(float(jn-(nseal+3))+0.5)/float(jec-jbc+1)
        if(jn.lt.nseal+2) y=wseal*(float(jn-2)+0.5)/float(nseal)
        if(jn.eq.0) y=0.
        if(jn.eq.nseal+2) y=wseal
        do 200 in=1,nnx
          x=tn+tg+lx*(float(in-6)+0.5)/float(iec-ibc+1)
          if(in.eq.1) x=0.
          if(in.eq.2) x=tn/2.
          if(in.eq.3) x=tn
          if(in.eq.4) x=tn + tg/2.
          if(in.eq.5) x=tn + tg
          if(in.eq.nnx) x=lx
          if(in.eq.nnx-1) x=lx - tn/2.
          if(in.eq.nnx-2) x=lx - tn
          if(in.eq.nnx-3) x=lx - tn - tg/2.
          if(in.eq.nnx-4) x=lx - tn - tg
          id=nnx*(jn-1)+in
          write(oput,25) it,id,iv,kc,n1,n2,n3,n4,n5
          write(oput,12) x,y,z
          write(oput,13) icf,gtype,ndf,config,cid,(pspc(iw),iw=1,6)
200     continue
100   continue
c  Element Data
      call head(it,id,iv,kc,n1,n2,n3,n4,n5)
      iv=4
      nodes=4
      n1=0
      kc=1 + (nodes+9)/10
      it=2
      config=1
      pid=0
      ceid=0
      t1=0.
      t2=0.
      t3=0.
      ney=nny-1
      nex=nnx-1
      do 400 jn=1,ney
        do 500 in=1,nex
          id=nex*(jn-1)+in
          write(oput,25) it,id,iv,kc,n1,n2,n3,n4,n5
          write(oput,22) nodes,config,pid,ceid,t1,t2,t3
          idv=nnx*(jn-1)+in
          write(oput,23) idv,idv+1,idv+nnx+1,idv+nnx
     &                   ,iz,iz,iz,iz,iz,iz
500     continue
400   continue
c  Close Neutral File
999   call head(it,id,iv,kc,n1,n2,n3,n4,n5)
      it=99
      kc=1
      write(oput,25) it,id,iv,kc,n1,n2,n3,n4,n5
99    continue
c     close(oput)
      return
      end
c
c \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c  This routine writes nodal temperature and x/y velocity data
c  to an output file in the neutral file format of PATRAN.  These
c  data can later be read by the PATRAN finite element graphics
c  i/o program.
c  Data are generated and written to the PATRAN neutral files for
c  the range of j from jb to jpate.
c  jpate must be less than jec (i.e., cavity cut off below upper
c  seal.
      subroutine notuv(t,u,v,ib,ie,jb,je,jpate,tww,twe,oput,
     &                   wseal,nseal,ibc,iec,jbc,jec,tn,tg,
     &                   kn,kg,con,lx,ly,
     &                   title,subt)
      real t(42,302)
      real u(42,302), v(42,302)
      real kn,kg,con,lx,ly
      integer oput
      character*14 neutfn
      character*80 title, subt(2)
      if(jpate.gt.je) goto 99
      if(jpate.lt.jb) goto 99
```

```
c     title='VSML temperature/velocity output to PATRAN'
c     subt(1) = 'Subtitle number 1'
c     subt(2) = 'Subtitle number 2'
70    format(a80)
71    format(i8,5e13.7)
72    format(2i9,e15.6,2i9)
c
      nny=jpate-jbc+nseal+3
      nnx=ie-ib+11
      diep=lx/float(iec-ibc+1)
      djnp=ly/float(jec-jbc+1)
      dyseal=wseal/nseal
c
      neutfn='tuvneut.pat'
c     open(unit=oput,file=neutfn,status='unknown')
c
c  Title Card
c
      write(oput,70) title
c
c  Summary Data
c
      nnodes=nnx*nny
      maxnod=0
      defmax=0.
      ndmax=0
      nwidth=3
      write(oput,72) nnodes,maxnod,defmax,ndmax,nwidth
c
c  Subtitle Cards
c
      write(oput,70) subt(1)
      write(oput,70) subt(2)
c
c  Calculate and write t/u/v data at each node
c
      call head(it,id,iv,kc,n1,n2,n3,n4,n5)
      do 100 jn=1,nny
        do 200 in=1,nnx
          id=nnx*(jn-1)+in
c  Calculate temperature at FEM node number id, tid
          i = ibc+ in - 6
          if(in.lt.5) i=ib + in - 2
          if(in.gt.nnx-5) i=iec + in - (nnx-5)
          j = jbc+ jn - (nseal+3)
          if(jn.lt.nseal+2) j = jb + jn - 2

          if(in.eq.1) then
            tid=tww
            goto 9
          endif
          if(in.eq.nnx) then
            tid=twe
            goto 9
```

```
endif

if(jn.eq.1) then
  tid=t(i,jb)
    if(in.eq.3) then
    cl=kn/tn
    cr=kg/tg
    den=cl+cr
    tl=t(ib,jb)
    tr=t(ib+1,jb)
    tid=((tl*cl)+(tr*cr))/den
    endif
     if(in.eq.5) then
     cl=kg/tg
     cr=kseal/diep
     den=cl+cr
     tl=t(ib+1,jb)
     tr=t(ibc,jb)
     tid=((tl*cl)+(tr*cr))/den
     endif
      if(in.eq.nnx-4) then
      cl=kseal/diep
      cr=kg/tg
      den=cl+cr
      tl=t(iec,jb)
      tr=t(iec+1,jb)
      tid=((tl*cl)+(tr*cr))/den
      endif
    if(in.eq.nnx-2) then
    cl=kg/tg
    cr=kn/tn
    den=cl+cr
    tl=t(iec+1,jb)
    tr=t(ie,jb)
    tid=((tl*cl)+(tr*cr))/den
    endif
  goto 9
endif

if((jn.gt.1).and.(jn.lt.nseal+2))then
  tid=t(i,j)
    if(in.eq.3) then
    cl=kn/tn
    cr=kg/tg
    den=cl+cr
    tl=t(ib,j)
    tr=t(ib+1,j)
    tid=((tl*cl)+(tr*cr))/den
    endif
     if(in.eq.5) then
     cl=kg/tg
     cr=kseal/diep
     den=cl+cr
     tl=t(ib+1,j)
```

```
          tr=t(ibc,j)
          tid=((tl*cl)+(tr*cr))/den
          endif
           if(in.eq.nnx-4) then
           cl=kseal/diep
           cr=kg/tg
           den=cl+cr
           tl=t(iec,j)
           tr=t(iec+1,j)
           tid=((tl*cl)+(tr*cr))/den
           endif
         if(in.eq.nnx-2) then
         cl=kg/tg
         cr=kn/tn
         den=cl+cr
         tl=t(iec+1,j)
         tr=t(ie,j)
         tid=((tl*cl)+(tr*cr))/den
         endif
       goto 9
     endif

     if(jn.gt.nseal+2) then
       tid=t(i,j)
         if(in.eq.3) then
         cl=kn/tn
         cr=kg/tg
         den=cl+cr
         tl=t(ib,j)
         tr=t(ib+1,j)
         tid=((tl*cl)+(tr*cr))/den
         endif
          if(in.eq.5) then
          cl=kg/tg
          cr=con/diep
          den=cl+cr
          tl=t(ib+1,j)
          tr=t(ibc,j)
          tid=((tl*cl)+(tr*cr))/den
          endif
           if(in.eq.nnx-4) then
           cl=con/diep
           cr=kg/tg
           den=cl+cr
           tl=t(iec,j)
           tr=t(iec+1,j)
           tid=((tl*cl)+(tr*cr))/den
           endif
         if(in.eq.nnx-2) then
         cl=kg/tg
         cr=kn/tn
         den=cl+cr
         tl=t(iec+1,j)
         tr=t(ie,j)
```

```
    tid=((tl*cl)+(tr*cr))/den
    endif
  goto 9
endif

if(jn.eq.nseal+2) then
    if(in.eq.2) then
    tid=(t(ib,jbc)+t(ib,jbc-1))/2.
    endif
     if(in.eq.3) then
     cl=kn/tn
     cr=kg/tg
     den=cl+cr
     tl=(t(ib,jbc)+t(ib,jbc-1))/2.
     tr=(t(ib+1,jbc)+t(ib+1,jbc-1))/2.
     tid=((tl*cl)+(tr*cr))/den
     endif
    if(in.eq.4) then
    tid=(t(ib+1,jbc)+t(ib+1,jbc-1))/2.
    endif
     if(in.eq.5) then
     cl=kg/tg
     cr=con/diep
     den=cl+cr
     tl=t(ib+1,jbc)
     tr=t(ibc,jbc)
     tnear=((tl*cl)+(tr*cr))/den
     tl=t(ib+1,jbc+1)
     tr=t(ibc,jbc+1)
     tfar=((tl*cl)+(tr*cr))/den
     tid=tfar+1.5*(tnear-tfar)
     endif
    if((in.gt.5).and.(in.lt.nnx-4)) then
    cl=kseal/dyseal
    cr=con/djnp
    den=cl+cr
    tl=t(i,jbc-1)
    tr=t(i,jbc)
    tid=((tl*cl)+(tr*cr))/den
    endif`
     if(in.eq.nnx-4) then
     cl=con/diep
     cr=kg/tg
     den=cl+cr
     tl=t(iec,jbc)
     tr=t(ie-1,jbc)
     tnear=((tl*cl)+(tr*cr))/den
     tl=t(iec,jbc+1)
     tr=t(ie-1,jbc+1)
     tfar=((tl*cl)+(tr*cr))/den
     tid=tfar+1.5*(tnear-tfar)
     endif
    if(in.eq.nnx-3) then
    tid=(t(ie-1,jbc)+t(ie-1,jbc-1))/2.
```

```
             endif
              if(in.eq.nnx-2) then
              cr=kn/tn
              cl=kg/tg
              den=cl+cr
              tl=(t(ie-1,jbc)+t(ie-1,jbc-1))/2.
              tr=(t(ie,jbc)+t(ie,jbc-1))/2.
              tid=((tl*cl)+(tr*cr))/den
              endif
             if(in.eq.nnx-1) then
             tid=(t(ie,jbc)+t(ie,jbc-1))/2.
             endif
           goto 9
         endif

c  Calculate x and y velocities at FEM node number id, xvel and yvel
9        if(in.le.5) then
           xvel=0.
           yvel=0.
           goto 8
         endif
         if(in.ge.nnx-4) then
           xvel=0.
           yvel=0.
           goto 8
         endif
         if(jn.le.nseal+2) then
           xvel=0.
           yvel=0.
           goto 8
         endif
         xvel=0.5*(u(i,j)+u(i-1,j))
         yvel=0.5*(v(i,j)+v(i,j-1))
8        write(oput,71) id,tid,xvel,yvel
200     continue
100   continue
99    continue
c     close(oput)
      return
      end
C \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
c
      subroutine head(it,id,iv,kc,n1,n2,n3,n4,n5)
      it=0
      id=0
      iv=0
      kc=0
      n1=0
      n2=0
      n3=0
      n4=0
      n5=0
      return
      end
```

The following file (igurun.com) was used to run IGU-2D in batch mode on the WATSUN micro-VAX.

```
$DEFINE FOR007 igu.plt
$DEFINE FOR009 igu.in
$DEFINE FOR004 igu.out
$DEFINE FOR003 nodeigu.pat
$DEFINE FOR008 tuvigu.pat
$DEFINE FOR014 out.tuv
$DEFINE FOR015 in.tuv
$DEFINE FOR017 igu.sfn
$DEFINE FOR018 igu.toy
$DEFINE FOR019 igu.qoy
$DEFINE FOR020 igu.qep
$run igu2
```